

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Máster Universitario en Investigación e
Innovación en TIC

TRABAJO FIN DE MÁSTER

**ESTUDIO DE TÉCNICAS DE
ANÁLISIS Y TRATAMIENTO DE
INFORMACIÓN EN P300:
APROXIMACIONES HACIA LA
CLASIFICACIÓN ADAPTATIVA**

Autor: Guillermo Sarasa Durán

Tutor: Francisco de Borja Rodríguez Ortiz

Septiembre 2016

ESTUDIO DE TÉCNICAS DE ANÁLISIS Y TRATAMIENTO DE INFORMACIÓN EN P300: APROXIMACIONES HACIA LA CLASIFICACIÓN ADAPTATIVA

Autor: Guillermo Sarasa Durán
Tutor: Francisco de Borja Rodríguez Ortiz

Grupo de Neurocomputación Biológica
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid

Septiembre 2016

Resumen

Las interfaces cerebro máquina (Brain Computer Interface, BCI) constituyen un acceso alternativo a las tecnologías de la información, mediante el registro y clasificación de señales procedente de la actividad cerebral. Las tecnologías de la información, abarcan cualquier sistema, servicio, o dispositivo informático. Como cualquier otra Interfaz de Usuario, los sistemas BCI representan un intermediario entre el software principal y el usuario. En este caso, este tipo de interfaz no proporciona un entorno amigable para el usuario de cara al acceso de la funcionalidad final. En su lugar, este intenta superar discapacidades físicas inherentes a la persona, aunque también es posible usarlo con otros fines, como científicos o meramente lúdicos. Controlar un sistema de control remoto con el uso simple de la voluntad, o comunicarse solo prestando atención a una matriz de deletreo, son posibles gracias al uso de sistemas BCI. No obstante, clasificar un proceso oculto en la actividad cerebral necesita de la aplicación de un complejo sistema de entrenamiento y tratamiento de la señal registrada. Las diferencias entre dos personas (conocido como diferencias intrasujeto), o entre una misma persona en el tiempo (intersujeto), hacen de este problema relativamente imposible de generalizar, y por tanto, muy difícil de clasificar. Y más teniendo en cuenta que las soluciones de aprendizaje automático, suelen estar enfocadas entorno a un entrenamiento inicial. Una de las soluciones para abordar este problema es el desarrollo de sistemas adaptativos, enfocados en reorientar sus parámetros de clasificación en función de la evolución de la fuente de datos. En los sistemas BCIs más comunes, se hace uso de la técnica de adquisición de datos por electroencefalograma. Esta señal, manifiesta una gran cantidad de ruido por naturaleza y, más importante, no conserva un comportamiento estacionario. No obstante, es plausible presuponer que a periodos de tiempo pequeños, existe un comportamiento suficientemente estacionario para su estudio.

En este proyecto, se han revisado un conjunto de técnicas usuales en el contexto de las BCIs con el objetivo de desarrollar una primera aproximación de sistema de clasificación con esta connotación de adaptabilidad. El paradigma de BCI escogido fue el *oddball*, sobre la señal de control P300. Partiendo de la hipótesis de la existencia de una estacionaridad local, se desarrolló un conjunto de sistemas para capturar la estructura fundamental. Comprendiendo esta las características instantáneas de la señal, conjunto con su evolución. En primera instancia, se implementó un sistema para medir la similaridad de la información, haciendo uso de métodos basados en la similaridad entre eventos en el tiempo. Del mismo modo, se trató de encontrar la configuración idónea de datos para maximizar la diferencia entre clases de eventos. Los métodos aplicados fueron: Correlación, Synchronization Likelihood y Normalized Compression Distance. Para poder medir las diferencias entre clases de eventos, se hizo del método de Minimum Quartet Tree Cost para su representación, y el coeficiente de Silhouette para el cálculo de distancia.

Para terminar, se desarrolló un clasificador sencillo sobre un conjunto de datos simple para poder verificar la capacidad de los resultados obtenidos. La capacidad del clasificador se desarrolló para mantener unos mínimos de funcionamiento, pero dejando un margen de mejora. Paralelamente, se obtuvo una puntuación para cada electrodo mediante el sistema ya explicado, y con ella se limitó la cantidad de datos del clasificador. Primero para aquellas veinte fuentes

mejores, seguido de las veinte peores, según la puntuación obtenida. Los resultados de esta modificación, mejoraron la calidad de la clasificación y, por extensión, la adaptación de los métodos aplicados.

Palabras Clave

Interfaz Cerebro Máquina, EEG, P300, Adaptación, Compresores, NCD, Correlación, Synchronization Likelihood, BCI Competition II.

Abstract

The Brain Computer Interfaces (BCI) are an alternative way to access to the information technologies, with the recording and classification of brainwave signals. Information technologies, includes any system, service, or IT device. Like any other User Interface, a BCI system represents an intermediary between the front-end software system and the user. In this case, this type of interface does not provide a user-friendly access to the final functionality environment. Instead, it tries to overcome physical disabilities inherent to the person, but it is also useful for other purposes such as scientific or merely playful. Control a remote control system with the simple use of the will, or communicate only paying attention to an speller matrix are possible through the use of BCI systems. However, brain activity hidden process classification requires the application of a complex training system and processing the recorded signal. The differences between two people (known as intra-subject differences), or from the same person over time (intersubject), make relatively impossible to generalize the problem, and therefore very difficult to classify. And even more due to machine learning systems usually only uses an initial training data set. One viable solution is the development of adaptive systems, focused on orientate its classification parameters depending on the evolution of the data source. Usually the data is recorded using the electroencefalographic (EEG) data acquisition technic. This signal shows a lot of noise and retains a non stationary behavior. However, it is plausible to assume that in small periods of time, there is a sufficiently stationary behavior for study.

In this project, a set of common BCI application techniques were reviewed, in order to develop an approximation of an adaptability classification system. The oddball paradigm was chosen to study the P300 characteristics. Assuming the existence a locally stationarity, a set of systems was developed to capture the principal structure. Understanding the structure like the instantaneous signal characteristics and evolution. In the first place, a system was implemented to measure the similarity of information. This system used similarity-based-on methods between events at different times. Similarly, right configuration data was tried to find to maximize the difference between classes of events. The methods used were: Correlation, Synchronization Likelihood and Normalized Compression Distance. In order to measure the differences between classes of events, the Minimum Cost Tree Quartet method was used for representation, and Silhouette coefficient distance measure.

Finally, a simple classifier was developed to verify the capability of the results obtained. The classifier was developed to accomplish a minimum classification rates, but leaving room for improvement. In parallel, a score for each electrode was obtained across the Silhouette coefficient. And then, the amount of data given to the classifier was limited to the top twenty electrodes, followed by the last twenty. The results of this modification, improved the quality of the classification and, by extension, the adaptation of the methods applied.

Key words

Brain Computer Interface, EEG, P300, Adaptation, Compressor, NCD, Correlation, Synchronization Likelihood, BCI Competition II

Agradecimientos

A mi tutor Francisco, por su continua ayuda en el desarrollo de este proyecto, y la gran paciencia y dedicación que ha aportado, tanto en el desarrollo de este proyecto, como en mi formación en general.

A todas aquellas personas que conviven la universidad. Que me han transmitido el gusto por conocer o me han facilitado el camino para encontrarlo. Y destacar a aquellas que se preocupan porque el sistema educativo sea algo que de gusto disfrutar.

A mis compañeros, Aaron y a Vinicio, por su ayuda, apoyo y amistad. Pero sobretodo, por acompañarme en mis procastinantes discusiones y mantener el aire acondicionado, apesar del fresquito.

A mis amigos por haberme ayudado, compartido tan buenos ratos, y conformado los ruedines de este y muchos otros logros más. Y en especial a Jaime, Pablo, Sergio, Alex, Javi y Molins, por su apoyo, tiempo y, sobre todo, paciencia.

Y para terminar, a mi gran familia. Desde aquellos que lloran por no probar las verduras, hasta los que aun resuenan en nuestras memorias.

Índice general

Índice de Figuras	XI
Índice de Tablas	XVI
1. Introducción	1
1.1. Motivación del proyecto	2
1.2. Objetivos y enfoque	3
2. Interfaces cerebro máquina. Estado del arte	5
2.1. Sistemas BCI	5
2.1.1. Técnicas de adquisición de señal	5
2.1.2. Tipos de señales de control	7
2.1.3. Oddball Paradigm: Matriz de deletreo	12
2.2. Técnicas de tratamiento de la señal	13
2.2.1. Filtrado	13
2.2.2. Extracción de Información: Aplicaciones comunes en EEG	14
2.2.3. Extracción de Información: Otras técnicas	17
2.3. Sistemas de clasificación	21
2.3.1. Análisis de Discriminante Lineal (LDA)	21
2.3.2. Máquina de Soporte Vectorial (SVM)	22
3. Metodología	25
3.1. Entorno de trabajo	25
3.1.1. Estructura de datos	25
3.1.2. Métodos de análisis usados	26
3.1.3. Sistemas de similaridad	27
3.1.4. Sistema de clasificación	31
3.2. Técnicas de análisis de información	32
3.2.1. Synchronization Likelihood (SL)	32
3.2.2. Distancia normalizada por compresor (NCD)	33
3.3. Simulaciones	34

3.3.1.	Construcción de objetos	34
3.3.2.	Cálculo de distancias y dendogramas por conjuntos	34
3.3.3.	Búsqueda paramétrica: Coeficiente de Silhouette	35
3.3.4.	Representación de resultados	35
3.4.	Conjuntos de Datos	35
3.4.1.	Conjunto de Datos 1: BCI Competition II	36
4.	Experimentos Realizados y Resultados	39
4.1.	Pruebas de validación: Ganando intuición	40
4.2.	Simulaciones sobre parámetros	40
4.2.1.	Estructura de las simulaciones	41
4.2.2.	Escenarios de pruebas	42
4.2.3.	Simulaciones sobre parámetros de estructura del evento	42
4.2.4.	Puntuación de electrodos	53
4.3.	Clasificación	54
4.3.1.	Clasificador simple	54
4.3.2.	Clasificador modificado	55
5.	Conclusiones y trabajo futuro	57
	Glosario de acrónimos	59
	Bibliografía	60
A.	Resultados de las simulaciones	65
A.1.	Simulaciones sobre parámetros de estructura	65
A.1.1.	Bzlib: conjunto A, subconjuntos del 1 al 5	66
A.1.2.	Bzlib: conjunto A, subconjuntos del 1 al 5 por invasión	71
A.1.3.	Zlib: conjunto A, subconjuntos del 1 al 5	76
A.1.4.	Zlib: conjunto A, subconjuntos del 1 al 5 por invasión	81
A.1.5.	Mx: conjunto A, subconjuntos del 1 al 5	86
A.1.6.	Mx: conjunto A, subconjuntos del 1 al 5 por invasión	91
A.1.7.	Std: conjunto A, subconjuntos del 1 al 5	96
A.1.8.	Std: conjunto A, subconjuntos del 1 al 5 por invasión	101
A.1.9.	Sl: conjunto A, subconjuntos del 1 al 5	106
A.1.10.	Sl: conjunto A, subconjuntos del 1 al 5 por invasión	111

Índice de Figuras

2.1. Esquema del proceso de un sistema BCI, previo a la comunicación con el sistema a controlar	5
2.2. Sistema 10 20 de posicionamiento de electrodos. Imagen adaptada de: https://upload.wikimedia.org/wikipedia/commons/2/20/system_for_EEG.svg	6
2.3. Ejemplo de Sistema de estimulación VEP. Cada estímulo parpadea a una frecuencia distinta, y el objetivo centra su atención en uno en particular. En este caso el cuadro superior.	8
2.4. Resultados de la toma de P300 sobre una muestra de personas de diferente edad. a) Amplitud del incremento en la señal, sobre la muestra. b) Latencia entre la muestra del estímulo y la aparición del P300 sobre la muestra. Adaptada de: https://upload.wikimedia.org/wikipedia/commons/1/11/P300_latency_and_amplitude_trajectories_sectional_dataset.png	9
2.5. Ejemplo ideal de P300. En el se muestran otros eventos, de menor amplitud, presentes en el evento. Los N representan los potenciales negativos, y los P los positivos. No obstante, estos eventos adicionales no suelen ser relevantes en las distintas implementaciones debido a su dificultad de detección. Imagen adaptada de: https://en.wikipedia.org/wiki/Event-related_potential	10
2.6. Esquema de la relación información/coste de P300, SSVEPS y SCP. Esquema adaptado de [1]	11
2.7. Ejemplo ideal de un fragmento de época de carácter. La linea continúa representa la señal registrada mientras que la discontinua muestra los estímulos presentados	12
2.8. Esquema de los diferentes niveles de definición de los datos. De menor a mayor: Frase, Palabra y Letra, que se conforma por estímulos a clasificar. El último nivel se representa en la figura 2.7	12
2.9. Matriz de deletreo usada comunmente en el paradigma <i>oddball</i>	13
2.10. Esquema de representación de la reconstrucción de la señal por estímulos usado en el proyecto	15
2.11. Esquema de representación de PCA. Adaptado de: http://phdthesis-bioinformatics-maxplanckinstitute-molecularplantphys.matthias-scholz.de/fig_pca_illu3d.png	16
2.12. Esquema de representación de la frontera de decisión en una SVM. Adaptado de: https://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png	
3.1. Esquema de la construcción de los datos. La señal se procesa como en la figura 2.10, para obtener las ventanas asociadas a evento o estímulo. Cada conjunto de ventanas produce una serie, y finalmente esta se concatena a otras series produciendo la fuente final a comparar.	28

3.2. Ejemplo del sistema de representación, mediante grafo. Se incluyen los coeficientes de calidad de cluster totales y parciales. Los nodos hoja coloreados en azul, representan a la clase asociada al evento P300.	30
3.3. Sistema de representación de resultados, mediante mapa de color. Los ejes de las matrices representan los valores de configuración como se define en 3.3.1. a) La primera matriz representa el coeficiente de Silhouette obtenido. b) La segunda representa la varianza asociada a cada valor	36
4.1. Mapas de color resultado de la prueba I (bzlib), sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)	43
4.2. Ejemplo de dendograma obtenido en la simulación sobre el primer conjunto de datos con el método bzlib.	45
4.3. Mapas de color resultado de la prueba II, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)	46
4.4. Ejemplo de dendograma obtenido en la simulación sobre el primer conjunto de datos con el método zlib.	47
4.5. Mapas de color resultado de la prueba III, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)	48
4.6. Ejemplo de dendograma obtenido en la simulación sobre el primer conjunto de datos con el método de máxima correlación.	49
4.7. Mapas de color resultado de la prueba IV, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)	50
4.8. Ejemplo de dendograma obtenido en la simulación sobre el primer conjunto de datos con el método de correlación estadística.	50
4.9. Mapas de color resultado de la prueba V, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)	51
4.10. Ejemplo de dendograma obtenido en la simulación sobre el primer conjunto de datos con el método SL.	52
4.11. Otro ejemplo de dendograma obtenido con SL. En este caso se observa la anomalía producida por la baja adaptación del método a la estructura del evento	52
4.12. Puntuaciones asignadas a cada electrodo mediante NCD con Bzlib	53
A.1. Mapa de color del subconjunto 1 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	66
A.2. Mapa de color del subconjunto 2 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	67
A.3. Mapa de color del subconjunto 3 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	68
A.4. Mapa de color del subconjunto 4 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	69

A.5. Mapa de color del subconjunto 5 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	70
A.6. Mapa de color del subconjunto 1 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	71
A.7. Mapa de color del subconjunto 2 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	72
A.8. Mapa de color del subconjunto 3 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	73
A.9. Mapa de color del subconjunto 4 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	74
A.10. Mapa de color del subconjunto 5 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	75
A.11. Mapa de color del subconjunto 1 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	76
A.12. Mapa de color del subconjunto 2 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	77
A.13. Mapa de color del subconjunto 3 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	78
A.14. Mapa de color del subconjunto 4 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	79
A.15. Mapa de color del subconjunto 5 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	80
A.16. Mapa de color del subconjunto 1 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	81
A.17. Mapa de color del subconjunto 2 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	82
A.18. Mapa de color del subconjunto 3 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	83
A.19. Mapa de color del subconjunto 4 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	84
A.20. Mapa de color del subconjunto 5 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	85
A.21. Mapa de color del subconjunto 1 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	86
A.22. Mapa de color del subconjunto 2 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	87
A.23. Mapa de color del subconjunto 3 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	88
A.24. Mapa de color del subconjunto 4 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	89
A.25. Mapa de color del subconjunto 5 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	90

A.26. Mapa de color del subconjunto 1 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	91
A.27. Mapa de color del subconjunto 2 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	92
A.28. Mapa de color del subconjunto 3 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	93
A.29. Mapa de color del subconjunto 4 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	94
A.30. Mapa de color del subconjunto 5 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	95
A.31. Mapa de color del subconjunto 1 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	96
A.32. Mapa de color del subconjunto 2 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	97
A.33. Mapa de color del subconjunto 3 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	98
A.34. Mapa de color del subconjunto 4 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	99
A.35. Mapa de color del subconjunto 5 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	100
A.36. Mapa de color del subconjunto 1 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	101
A.37. Mapa de color del subconjunto 2 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	102
A.38. Mapa de color del subconjunto 3 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	103
A.39. Mapa de color del subconjunto 4 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	104
A.40. Mapa de color del subconjunto 5 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	105
A.41. Mapa de color del subconjunto 1 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	106
A.42. Mapa de color del subconjunto 2 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	107
A.43. Mapa de color del subconjunto 3 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	108
A.44. Mapa de color del subconjunto 4 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	109
A.45. Mapa de color del subconjunto 5 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	110
A.46. Mapa de color del subconjunto 1 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.	111

A.47. Mapa de color del subconjunto 2 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.	112
A.48. Mapa de color del subconjunto 3 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.	113
A.49. Mapa de color del subconjunto 4 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.	114
A.50. Mapa de color del subconjunto 5 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.	115

Índice de Tablas

3.1.	Tabla de resultados de la segunda competición de BCI	37
4.1.	Resultados obtenidos por los distintos métodos para cada una de las pruebas de validación realizadas. MaxCorr: máxima correlación, StdCorr: Correlación estadística, SL: Synchronization Likelihood, NCD: Distancia normalizada por compresor.	41
4.2.	Resultados máximo, mínimo y medio obtenidos para cada simulación sobre el primer conjunto de datos con Bzlib para cada subconjunto	44
4.3.	Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con Bzlib. A diferencia de la tabla 4.2, en esta tabla se muestran los coeficientes para cada clase por separado.	45
4.4.	Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con Zlib para cada conjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior	46
4.5.	Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con Zlib. A diferencia de la tabla 4.4, en esta tabla se muestran los coeficientes para cada clase por separado.	47
4.6.	Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con correlación máxima para cada subconjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior	48
4.7.	Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con máxima correlación. A diferencia de la tabla 4.6, en esta tabla se muestran los coeficientes para cada clase por separado.	48
4.8.	Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con máxima correlación para cada subconjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior	49
4.9.	Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con correlación estadística. A diferencia de la tabla 4.8, en esta tabla se muestran los coeficientes para cada clase por separado.	49
4.10.	Resultados máximo, mínimo y medio de clusterización total por cada simulación de cada subconjunto del primer conjunto de datos con SL. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior	51
4.11.	Resultados parciales obtenidos durante la simulaciones del primer conjunto de datos con SL. A diferencia de la tabla 4.10, en esta tabla se muestran los coeficientes para cada clase por separado.	51
4.12.	Resultados de la clasificación del conjunto de datos introducido en 3.4.1, para cada sesión.	55

4.13. Resultados de la clasificación del conjunto de datos introducido en 3.4.1, para cada configuración de electrodos de acuerdo con los obtenidos en la simulación	
4.2.4	55

1

Introducción

Hoy en día, no es posible concebir el uso de un ordenador personal o de un dispositivo móvil sin un complemento gráfico que facilite su uso al usuario final. De nada sirve poder establecer comunicación con un servicio remoto, o interactuar con un sistema de altas prestaciones si no se tiene la capacidad, o conocimiento, para poder interoperar con este. La interfaz de usuario gráfica, o GUI del inglés, permite acercar parte de la funcionalidad de un modo sencillo e intuitivo. Acceder al correo electrónico sin conexión a internet o almacenar remotamente contenido con la interoperabilidad de un sistema local, son ejemplos de interfaces desarrolladas para hacer usable y, por extensión, accesible la tecnología a cualquier usuario. Por tanto, las Interfaces de Usuarios, o UI (User interface), se han convertido en un ente tan importante como la propia funcionalidad del conjunto.

Un ejemplo menos usual de UI, y más restringido, es el de las Interfaces Cerebro Computador (BCI, Brain Computer Interface) o Interfaces Cerebro Máquina (BMI, Brain Machine Interface). Este tipo de interfaz tiene como objetivo, al contrario que los ejemplos de UI mencionados previamente, proveer de capacidades a personas, normalmente en situaciones de discapacidad. Es decir, en lugar de hacer de vía de comunicación entre la funcionalidad y el usuario, ser la propia funcionalidad en si misma, capacitando al usuario a realizar tareas, en algunos casos, con el propio mundo real.

Las BCIs, por tanto, representan una definición de UI interesante no solo por su naturaleza de UI, sino por las necesidades existentes en nuestra sociedad hacia estas. Del mismo modo, resultan unos sistemas interesantes de cara al ámbito investigador, debido a que su fuente de datos de entrada permite el estudio de procesamiento de información cerebral, así como caracterizar ciertos estados mentales a través de estas señales. Tanto la investigación del comportamiento en si, como su estructura e información para poder diferenciarlo, son algunos de los campos de investigación presentes en la rama de conocimiento entorno a las BCIs. De hecho, en los últimos años la cantidad de publicaciones sobre BCIs ha crecido de manera exponencial.

Una de las razones causantes de este crecimiento es la dificultad inherente a la implementación de sistemas BCI. Mientras las diferentes técnicas de adquisición de datos intentan maximizar la calidad de estos minimizando la invasión en el sujeto, y por tanto, los riesgos del sistema; los sistemas de procesamiento de información tratan de apaliar este equilibrio. El cual resulta descompensado por el alto coste (tanto espacial, como económico) de los sistemas con mayor calidad de datos. Lo que deja mayor peso en el desarrollo de metodologías de análisis y tratamiento de información que compensen este hecho. Un ejemplo de los esfuerzos por mejorar estos

sistemas, son las competiciones de BCI. Estas proporcionan una reproducibilidad de resultados y comparación de métodos de clasificación y filtrado, debido a la fuente de datos común.

Estos problemas constituyen la toma de contacto para el desarrollo de un sistema BCI. No obstante, el mantenimiento de métodos particulares resulta más costoso, debido a la variabilidad [2] de un sistema complejo que conforma un sistema vivo. Esta variabilidad se manifiesta de dos modos igual de problemáticos: Entre distintos sujetos y entre diferentes estados de un mismo sujeto. Este último, resulta de la propia no estacionariedad de los datos obtenidos. La latencia del evento, o la amplitud [3] de este son algunos de los ejemplos cambiantes entre distintos sujetos. Aunque, por supuesto también existen variaciones notablemente diferenciables entre muestras de personas, por ejemplo genéticos [4] así como por enfermedades [5], o incluso personalidad [6].

Por tanto, tras solventar los problemas inherentes a la señal, es necesario enfocar el desarrollo de estos sistemas en ser resistentes ante estas derivas intra- y inter-sujeto. Una alternativa a sistemas genéricos con menor precisión, es el desarrollo de sistemas adaptativos a cada sujeto, e incluso a la propia situación de este.

El trabajo desarrollado en este Trabajo Fin de Máster, se ha enfocado a la revisión y aplicación de diferentes técnicas de análisis y tratamiento de información, como primera aproximación hacia el desarrollo de un sistema adaptativo con esta connotación. Más concretamente, en el contexto del paradigma *oddball* sobre la señal de control P300. La elección de esta señal de control viene motivada tanto por su relativa simplicidad de detección a simple vista, y la facilidad de realización de la toma de datos de cara al sujeto, comparada con otras señales de control.

1.1. Motivación del proyecto

La motivación principal de este proyecto, es la de a través de diferentes métodos obtener adaptación ante problemas de tipo BCI. Entendiendo adaptación como todo aquel proceso que permite hacer a un sistema más preciso en función de parámetros del usuario, y no del sistema en si. Es decir, en lugar de un sistema que mantenga una precisión de manera genérica, un sistema que se especialice a la toma de datos, en este caso el usuario. Algunas soluciones BCI también se han enfocado al desarrollo de sistemas adaptativos [7] [8] en lugar de sistemas con mayor capacidad [9] debido a la vertiente de eficacia que significa el desarrollo de adaptación. Otras implementaciones, por el contrario, optan por sistemas genéricos enfocados a realizar entrenamientos periódicos [10] entre sujetos, y en algunos casos costosos, para solventar la no estacionariedad inherentes al problema. Y aunque este tipo de procedimientos dan resultados relativamente positivos, los costes del entrenamiento continuo y su fragilidad ante cambios en el estado de la actividad, hacen a la adaptación una opción más interesante.

El tipo de señal de adquisición, electroencefalograma (EEG), así como la señal de control P300, son escogidas por sus características de simplicidad y facilidad de acceso. Atributos necesarios dado el enfoque principal de este tipo de sistemas. Además, los problemas existentes entorno a estos, son dificultades atractivas desde el punto de vista de los sistemas adaptativos en tiempo real. La capacidad de adaptación de estos al usuario, requiere de la detección, en parte, del comportamiento o estructura de la actividad. Lo cual, podría además aportar información o reducir la incertidumbre entorno a otras investigaciones relacionadas con el campo de estudio, más allá de solo resolver problemas en el entorno de los sistemas BCI.

La adaptación entorno a este tipo de problemas, requiere de un acercamiento a alguna propiedad o atributo único de cada sujeto, pero modelizable o extraíble de forma general. En otras palabras, una métrica, o conjunto de ellas, diferenciable y un método que permita extraerla. En este proyecto, la adquisición de estas variables se fundamenta en la hipótesis de que la estructura del P300 es única, o al menos, diferenciable. De este modo, la extracción de la información, al

menos parcial, de esta debería de constituir una base para el desarrollo de una futura métrica de adaptación.

Para terminar, este proyecto también está motivado por la realización de un proyecto de tesis doctoral, enfocado en la obtención de adaptación a través del análisis de la información contenida en diversas fuentes de datos. Por lo que, los trabajos futuros o posibles continuaciones sobre algunos de los resultados obtenidos de este proyecto, serán realizados a continuación durante el desarrollo de este proyecto.

1.2. Objetivos y enfoque

El enfoque de este proyecto, como se ha adelantado en la introducción, es el de revisar e investigar entorno a diferentes sistemas de análisis y tratamiento de información como aproximaciones hacia la realización futura de un sistema de clasificación adaptativa. Para ello se han establecido los siguientes objetivos:

1. **Revisión de técnicas de análisis y tratamiento de información**, con el objetivo de familiarizarse con algunas de los procedimientos usados en soluciones BCI previas; y conocer algunos métodos en actual desarrollo.
2. **Selección de conjunto de datos**, Debido a la originalidad del sistema de análisis seleccionado, será necesario probar el conjunto con una fuente de datos fiable y tratada para reducir los problemas derivados de este.
3. **Implementación de un entorno de desarrollo**, para unificar los diferentes sistemas y obtener un ambiente de pruebas común. Y del mismo modo, minimizar los problemas derivados del desarrollo de software. Por ejemplo errores arrastrados en simulaciones grandes u operaciones delicadas de alto coste.
4. **Desarrollo de pruebas sobre los métodos implementados**, que permitan verificar la capacidad de los distintos métodos, y comparar y contrastar los resultados obtenidos frente a la fuente de datos.
5. **Aplicación y contraste de los datos obtenidos sobre un clasificador real**, para medir de forma empírica el efecto parcial de la información obtenida por el sistema, sobre un problema real. En este caso mejorar un clasificador sobre un conjunto de datos.
6. **Elaboración de un informe y discusión de resultados sobre el trabajo realizado**, como conclusión sobre el conjunto del proyecto.

2

Interfaces cerebro máquina. Estado del arte

2.1. Sistemas BCI

Un sistema BCI se compone por cuatro funciones principales: Adquisición de señal, Filtrado de eventos (Señal de control), y Entrenamiento y Clasificación (Procesamiento de información). Este esquema se representa en la figura 2.1.

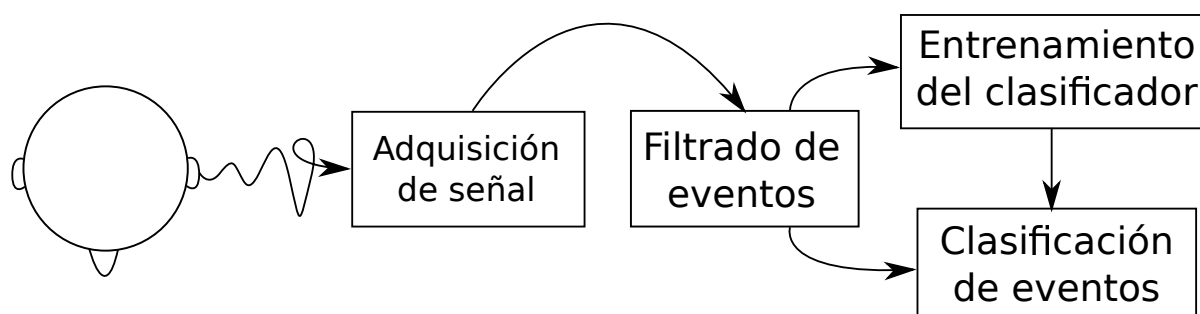


Figura 2.1: Esquema del proceso de un sistema BCI, previo a la comunicación con el sistema a controlar

2.1.1. Técnicas de adquisición de señal

El objetivo de un sistema BCI es el de obtener información de la actividad cerebral de una persona [11]. Para poder llevar a cabo este análisis, es necesario definir un método de registro de la actividad a monitorizar. Dependiendo del sistema de adquisición que se use para registrar la actividad, se podrán realizar unos análisis u otros. Esto se debe a las zonas de registro accesibles o la cantidad de ruido inherentes a los distintos sistemas de adquisición.

Las principales técnicas de adquisición de señal son las siguientes:

EEG (Electroencefalograma)

La adquisición de señal mediante electrodos repartidos en la superficie del cuero cabelludo, es definida como EEG. Este sistema registra los cambios de potencial provocados por la actividad cerebral, dando mayor importancia a aquellos procesos producidos en la superficie del cerebro. La distancia entre las fuentes de actividad y los electrodos también perjudica la calidad de la señal registrada [12]. El cabello, el cráneo o la piel son algunos de los factores que influyen en la calidad de la señal resultante [13].

Por otro lado, la accesibilidad que EEG proporciona es alta, debido a la facilidad de aplicación en comparación con otros sistemas de adquisición de señal [14]. La no invasividad del sistema, así como la gran resolución temporal, son algunas de las ventajas que apoyan su uso sobre el resto de técnicas de adquisición de señal. No obstante, el sistema dispone de una baja resolución espacial, debido a que solo es posible obtener datos detectables en la corteza del cráneo. Por lo que EEG, puede resultar insuficiente para otros tipos de análisis.

Un problema asociado a la variabilidad entre usuarios, es la dependencia del tamaño del cráneo del usuario. Este se compensa con el sistema 10-20 [15], que constituye un estándar de colocación de electrodos. La cantidad de electrodos útiles suele ser de 64, aunque en algunos sistemas BCI pueden usarse menos debido a las restricciones de la plataforma hardware, o una simplificación del problema. Este estándar define el mapa de electrodos de forma relativa a la distancia entre ciertos puntos del cráneo, lo que proporciona cierta generalidad a la hora de obtener datos entre sujetos.

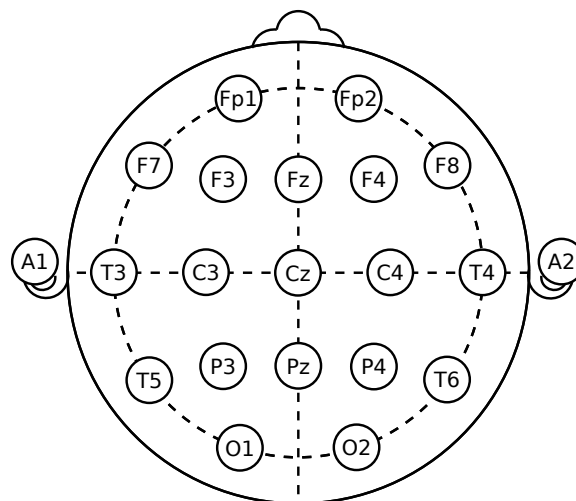


Figura 2.2: Sistema 10 20 de posicionamiento de electrodos. Imagen adaptada de: https://upload.wikimedia.org/wikipedia/commons/7/70/21_electrodes_of_International_10-20_system_for_EEG.svg

El tamaño de las plataformas que proporcionan capacidad para adquirir EEG es razonable, y tiene una facilidad de uso considerable. Del mismo modo, los costes de adquisición de las plataformas hardware, en comparación con el resto de sistemas, es mínimo. De este modo EEG constituye un sistema de adquisición de información con una capacidad relativa, debido a sus grandes limitaciones, pero razonables, dados sus costes.

MEG (Magnetoencefalograma)

Los sistemas de adquisición MEG, se centran en la misma actividad que EEG pero haciendo uso de campos magnéticos para medir las diferencias de potencial. El uso de campos magnéticos

proporciona una resolución espacial un poco mayor a EEG, sin perder resolución temporal [14]. No obstante el equipo necesario tiene unos costes espaciales y económicos mucho más altos a EEG. Especialmente en su transporte y manejo dado que debe de constituirse como un periférico de uso continuo por un usuario. Además, este tipo de sistemas requieren de personal especializado y por tanto, no son accesibles para un usuario final sin soporte constante [12].

fMRI (Imagen por Resonancia Magnética funcional)

Al contrario que los dos sistemas anteriores, fMRI registra la actividad cerebral midiendo las interacciones metabólicas del cerebro. Más concretamente, las concentraciones de oxígeno por zonas. Este tipo de técnica posee una resolución espacial muy alta comparada con las dos técnicas previamente descritas [14], [11]. Sin embargo, carece de una resolución en tiempo, fundamental en la implementación de sistemas BCI. Del mismo modo, los costes del equipo (Para producir el campo magnético necesario para poder medir las diferencias de oxígeno dentro del cerebro) hacen inviable esta opción para cualquier propósito fuera del ámbito de investigación.

ECoG (Electrocorticograma)

ECoG se centra en un registro de la actividad idéntico a EEG, con la diferencia de que los electrodos son insertados entre la corteza craneal y la superficie del cerebro. Está técnica elimina gran parte del ruido producido en EEG, y mejora la calidad de la señal adquirida [14], en general. No obstante, los riesgos de una operación invasiva o los costes de mantenimiento de un sistema de estas características, descartan esta técnica como técnica accesible para un sistema BCI.

Conclusión

En conclusión, EEG representa una técnica de adquisición de señal ruidosa, pero razonable dada la facilidad de uso y adquisición de la plataforma necesaria en comparación con el resto de alternativas.

2.1.2. Tipos de señales de control

El segundo componente fundamental de un sistema BCI es el patrón, o señal de control, a reconocer dentro de la actividad registrada. Debido a la cantidad de procesos inevitables dentro de la actividad cerebral, es necesario establecer un proceso que identifique una respuesta ante un comportamiento o estímulo [16]. Y de este modo, poder mapear las diferentes opciones dentro de la señal adquirida.

Dentro de las señales de control podemos diferenciar en primera instancia, aquellas como respuesta a estímulos externos, o evocadas; y aquellos producidos por la intención o voluntad del sujeto. En el caso de las evocadas, su aplicación es más reducida, debido a su componente no intencionada, pero tienen una dificultad menor de cara a su detección.

VEP (Potenciales Evocados Visuales)

La señal de control VEP, está definida en el dominio de frecuencias de la señal. Al contrario que otras señales de control, no se define como un comportamiento. Esta se define como una respuesta, no definida, a un estímulo repetido con cierto periodo. De forma que la manifestación de la respuesta, aunque desconocida, se mantenga en promedio a una determinada frecuencia

[7]. En función del espectro de frecuencias utilizado los VEPs se definen como TVEP (VEP transitorio), con frecuencias menores de 4Hz; o SSVEPs (Steady State VEP), con frecuencias superiores a los 4Hz [14], [11].

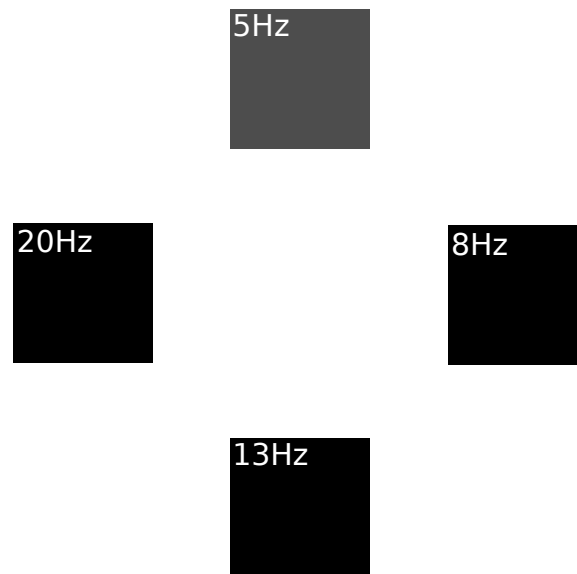


Figura 2.3: Ejemplo de Sistema de estimulación VEP. Cada estímulo parpadea a una frecuencia distinta, y el objetivo centra su atención en uno en particular. En este caso el cuadro superior.

Esta señal de control, permite diferenciar un determinado número de frecuencias por persona (dependiendo del sistema BCI en conjunto). La implementación más común, es la de permitir al usuario diferenciar entre un determinado número de opciones. Cada una de ellas parpadeando a una frecuencia determinada.

SCP (Potenciales Corticales Lentos)

La señal de control SCP, al contrario que la anterior, es una señal producida sin estímulo previo. Esta requiere de un entrenamiento costoso por parte del sujeto, para producir aumentos en el potencial de la corteza cerebral. La producción de estos incrementos de potencial, es lenta y muy dependiente del sujeto. Aunque la característica de ser producidos con la intencionalidad del sujeto es interesante de cara a un sistema BCI, el entrenamiento, lentitud de producción y limitación a una variable a graduar, hacen de SCP una señal de control con pocas aplicaciones reales.

P300

La señal de control P300, se define como la manifestación de un aumento leve de la señal en ciertas partes de la actividad cerebral [2]. Tras la presentación de un elemento infrecuente con respecto a un conjunto de estímulos (ya sean visuales, sonoros [17] o táctiles [18]) frecuentes, se produce este incremento en ciertas zonas del cerebro. Este aumento en el potencial de la señal, se manifiesta en torno a los 300ms, aunque al igual que la duración del mismo, este retardo estímulo-respuesta puede variar en gran medida [3]. En la figura 2.4 puede observarse la deriva de la amplitud y de la latencia media, medida en sobre una muestra de personas de distinta edad. El comportamiento ideal, obtenido de situaciones controladas, producto de múltiples promedios de esta señal puede observarse en la imagen 2.5

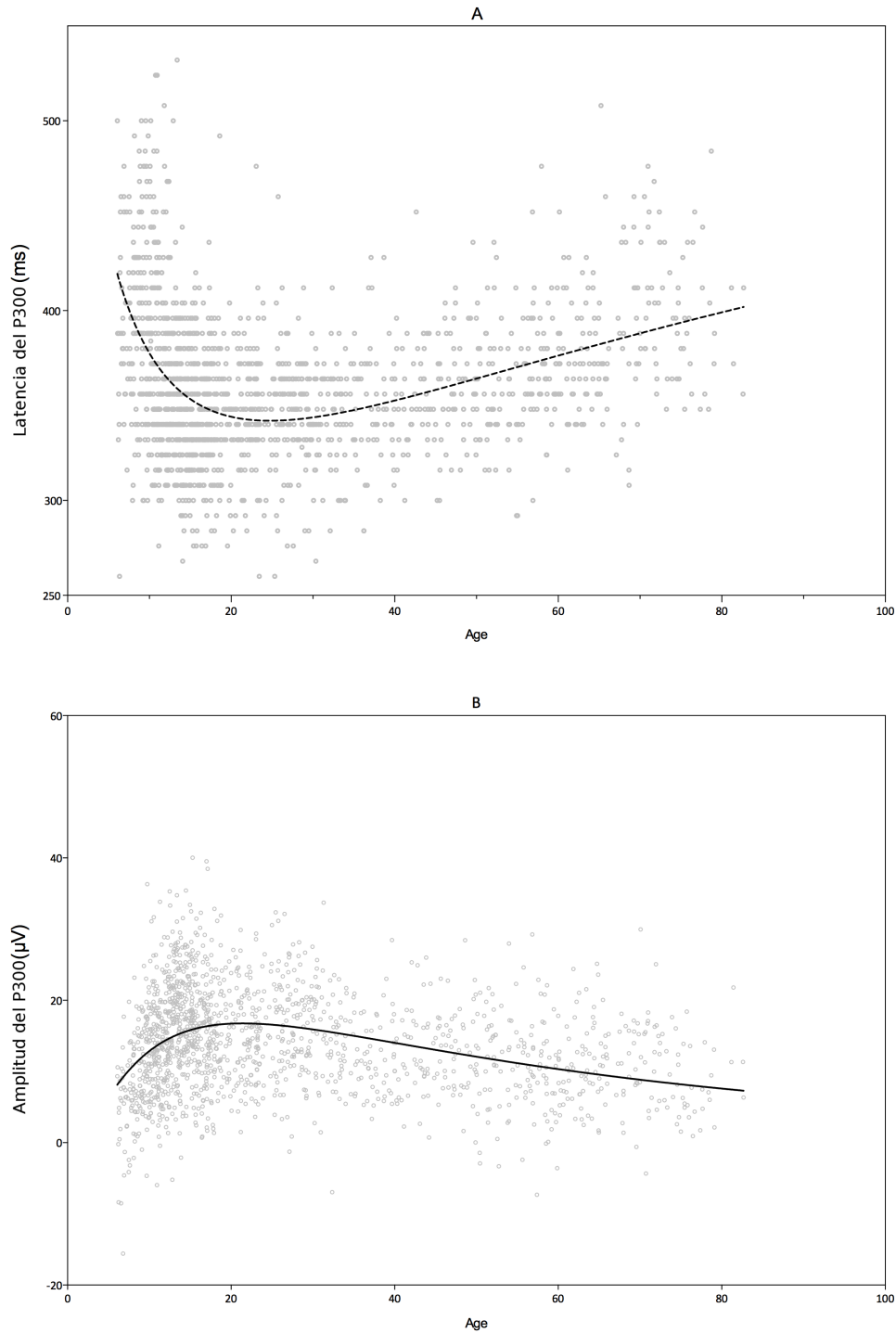


Figura 2.4: Resultados de la toma de P300 sobre una muestra de personas de diferente edad. a) Amplitud del incremento en la señal, sobre la muestra. b) Latencia entre la muestra del estímulo y la aparición del P300 sobre la muestra. Adaptada de: https://upload.wikimedia.org/wikipedia/commons/1/11/P300_latency_and_amplitude_trajectories_across_sectional_dataset.png

Esta señal de control se encuentra dentro de las evocadas, pero no dentro de VEPs, debido a que también puede ser producido por estímulos no visuales. La detección de esta señal, es compleja debido a que la relación existente entre el P300 y la respuesta ante un estímulo infrecuente no es directa [19]. Al contrario que los VEPs que se analizaban en frecuencia en un periodo amplio, P300 necesita del análisis de varios estímulos para, en promedio, observar el incremento en el potencial. Esta es una de las principales estrategias para reducir las componentes de incertidumbre de apariencia y de no estacionariedad de EEG. Por otro lado, el modelado de esta señal entre usuarios, o incluso en un mismo usuario no es posible debido a que este es distinto entre personas y cambia con el tiempo [3]. Por tanto, el aprendizaje de la señal debe ser individual de cada sesión, o por el contrario, usar métodos para capturar componentes de la señal lo suficientemente robustas a estos cambios.

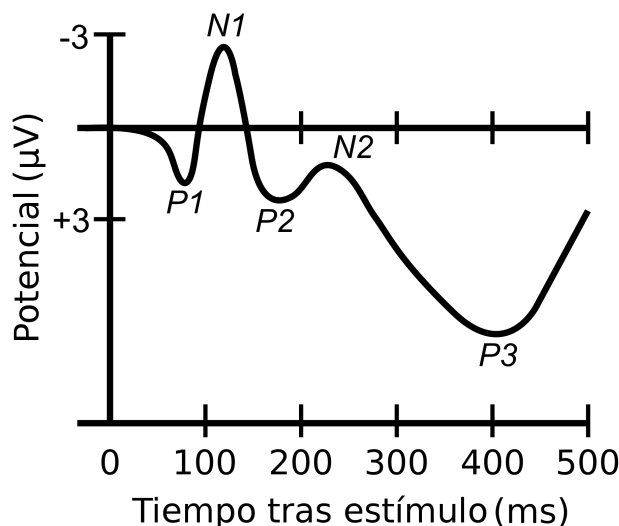


Figura 2.5: Ejemplo ideal de P300. En el se muestran otros eventos, de menor amplitud, presentes en el evento. Los N representan los potenciales negativos, y los P los positivos. No obstante, estos eventos adicionales no suelen ser relevantes en las distintas implementaciones debido a su dificultad de detección. Imagen adaptada de: https://en.wikipedia.org/wiki/Event-related_potential

Esta señal de control tiene una aplicación limitada al desarrollo de aplicaciones, que mediante "trucos", permiten obtener información. Es decir, la comunicación no es fruto de una interacción directa con el usuario. En su lugar es necesario aprovechar la capacidad del sistema BCI para comunicarse con el sujeto mediante conjuntos de estímulos, a diferenciar entre frecuentes e infrecuentes. En otras palabras, P300 solo permite diferenciar si el usuario reconoce, en promedio, un evento entre muchos, por lo que la capacidad de comunicación recae en el sistema BCI. Este paradigma, es denominado "oddball paradigm" [2], que se explica en profundidad en la sección 2.1.3. En este paradigma se presenta una serie de estímulos, y se intenta detectar aquel que resulta infrecuente para el sujeto, o en otras palabras, reconocible o relevante. Una de las muchas implementaciones de este problema, es mediante el uso de una matriz de deletreo, en el que se intensifican alternadamente filas y columnas. El objetivo es ser capaz de construir una palabra o conjunto de ellas, deletreando cada palabra caracter a caracter [20]. En este sistema, la aparición de los estímulos es del orden de décimas de segundo, por lo que el intervalo asociado al evento se superpone entre estímulos.

Aunque la velocidad de manifestación de esta señal es lo suficientemente corta como para implementar un sistema BCI, las repeticiones necesarias por el sistema de clasificación, así como la velocidad de respuesta de este, hacen de muchos sistemas con P300 excesivamente lentos.

Ritmos sensiomotores

Para terminar, los ritmos sensiomotores [21] son caracterizaciones en las bandas de frecuencia ante la voluntad de imaginar movimientos o acciones de combinaciones de estos. La detección de esta señal de control resulta bastante complicada, pero como contrapartida resulta mucho más útil de cara a su aplicación. Debido a que no requiere estímulos previos, y es mapeable a varios movimientos, la capacidad de explotación es alta. Por otro lado, la distancia de detección entre los distintos estímulos depende en gran medida del usuario.

La principal técnica usada para la detección de esta señal es "Band power" [22]. Esta técnica caracteriza la señal mediante un vector conformado por la media de cada banda de frecuencia de 1 Hz. Los resultados de este tipo de BCIs, no obstante, depende en gran medida de la calidad de la señal o la cantidad de entrenamiento realizado por el usuario. Y de este modo, la cantidad de movimientos mapeables por el sistema.

Conclusiones

De las señales de control revisadas, P300 representa una opción intermedia entre utilidad y accesibilidad [1]. Por un lado, permite hacer uso de sistemas con menor calidad y por tanto mayor ruido, debido a que su aplicación no necesita diferenciar entre distintas respuestas, como es el caso de los ritmos sensiomotores. La cantidad de entrenamiento necesario es, también, muy inferior comparado con los ritmos sensiomotores, y sensiblemente superior a la necesitada por VEPs [14]. De este modo, los costes dedicados a la plataforma de adquisición de señal pueden ser menores, y por tanto más accesibles a un usuario medio.

Por otro lado, la utilidad de P300 es bastante inferior a otros debido a su propia definición de señal evocada. Y aunque la velocidad de producción teórica es muy superior a la definición de otras como SCP o ritmos sensiomotores, esta se ve alterada por la calidad del conjunto de la BCI. Esto mismo se manifiesta en las señales de tipo VEPs, que a menor tiempo el ratio de información es mayor. No obstante, el método de estimulación así como la capacidad del usuario, resultan más problemáticos de cara a la capacidad de atención del sujeto. Y por tanto, a la calidad de la señal. De este modo, la utilidad de P300 resulta un punto intermedio en el conjunto de señales de control.

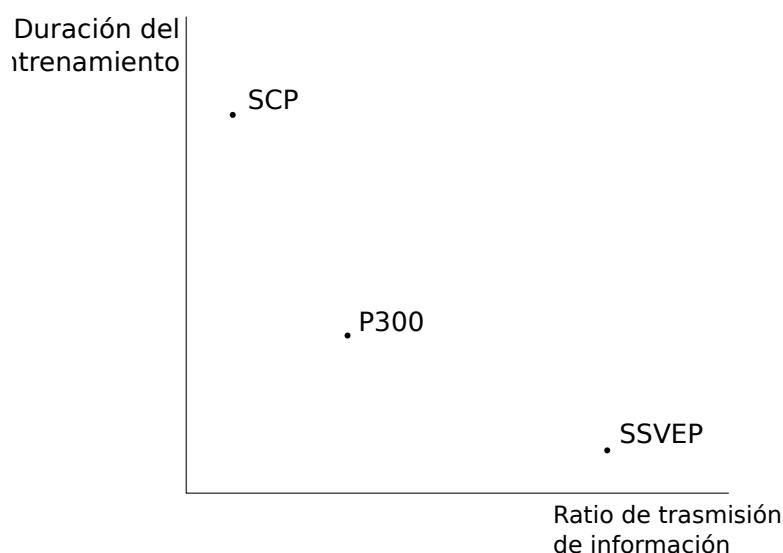


Figura 2.6: Esquema de la relación información/coste de P300, SSVEPS y SCP. Esquema adaptado de [1]

2.1.3. Oddball Paradigm: Matriz de deletreo

El paradigma "oddball", define un experimento en el que se muestra una serie de estímulos visuales, sonoros [17] o incluso táctiles [18], y se intenta adivinar cual es el deseado por el sujeto. Aplicado a EEG, el método aplicado para discriminar es mediante la detección de ERPs (potenciales relacionados con un evento). El ejemplo más común, en EEG, es el de la matriz de deletreo, comúnmente representada como se muestra en la figura 2.9, aunque existen otras versiones. En esta implementación del paradigma, se muestra una matriz con letras y números, y se invita al sujeto a prestar atención a uno. Acto seguido empiezan a parpadear subconjuntos de letras y números correspondientes a filas y columnas, con el objetivo de minimizar los estímulos a clasificar.

Usualmente, no se clasifica de forma directa en una toma de datos, sino que se realizan múltiples parpadeos de los mismos estímulos (Ejemplificado en la figura 2.7). La utilidad es la de realizar promedios, ya sea en la clasificación o en el tratamiento de la señal. La primera razón para esta repetición de estímulos es que no siempre se manifiesta el incremento de amplitud fruto del P300 [18]. Del mismo modo, el ruido y las variaciones fruto de la naturaleza de la señal, pueden contaminar los datos e impedir que se clasifiquen correctamente. Y aunque no resuelve completamente el problema, si permite reducir la cantidad de ruido. Al periodo completo de detección se le denomina "Época de caracter". De forma usual, los estímulos se dividen en dos conjuntos de seis (dos filas y dos columnas). Cada estímulo consta de una intensificación, de fila o columna, seguido de un descanso. Esto constituye la unidad atómica de la época de caracter, que se repite alternando entre estímulos.

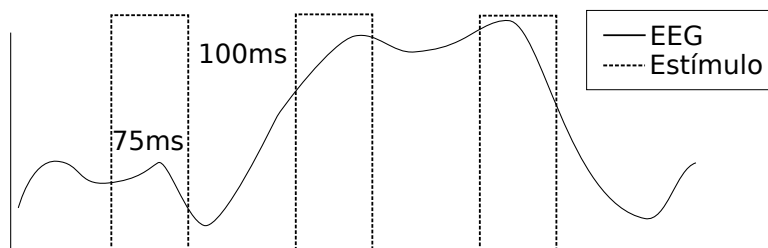


Figura 2.7: Ejemplo ideal de un fragmento de época de carácter. La línea continua representa la señal registrada mientras que la discontinua muestra los estímulos presentados

En la mayoría de BCIs que hacen uso de este tipo de implementación, el objetivo es el de deletrear palabras completas [19]. De este modo podemos diferenciar, como ayuda al concepto no estacionario de la señal, entre diferentes niveles de detección. El primer nivel, y más bajo, definido por distinguir la fila y columna objetivo; El segundo, que detecta las diferentes letras de una palabra; y finalmente el tercero, que detecta las palabras contenidas en una frase o prueba, en términos de competiciones de BCI.

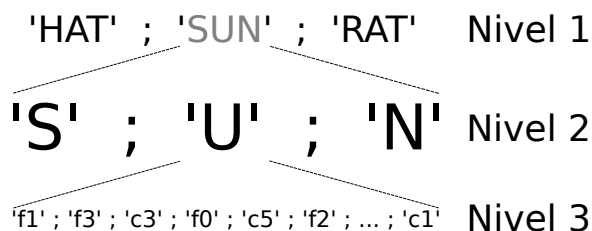


Figura 2.8: Esquema de los diferentes niveles de definición de los datos. De menor a mayor: Frase, Palabra y Letra, que se conforma por estímulos a clasificar. El último nivel se representa en la figura 2.7

Este paradigma posee unos problemas por su propia definición que lo hacen más complicado de cara a la clasificación final de los estímulos. El primero, como ya se ha comentado en la sección 2.1.2, es la no estacionaridad de la señal. Sesiones más largas, implicarán errores más altos. En segundo lugar, el ruido y no linealidad de la señal dificultan la generalización y filtro del evento correspondiente. Y en tercer lugar, la definición de acierto del clasificador. Para poder concluir que el clasificador ha acertado, es necesario dos éxitos consecutivos (fila y columna). Por lo que, cualquier clasificador verá reducido su tasa de éxito de inicio al cuadrado de la probabilidad de acierto ($P_{final}(x) = P(x)^2$).

A	B	C	H	I	J
D	E	F	G	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	0	1	2	3
9	4	8	5	6	7

Figura 2.9: Matriz de deletreo usada comunmente en el paradigma *oddball*

2.2. Técnicas de tratamiento de la señal

Partiendo del uso de la señal de adquisición EEG, y con el objetivo de detectar la señal de control P300, en esta sección se revisarán algunas de las técnicas con mejores resultados usadas para el tratamiento y análisis de la información de la señal.

2.2.1. Filtrado

Para poder realizar un análisis correcto de la información de la señal, es necesario previamente intentar minimizar las componentes no deseadas, o ruido, de la señal. En el caso de EEG, es especialmente necesario, debido a que la cantidad de procesos no deseados que son registrados es considerable [23]. Además, la discriminación individual de cada uno de ellos es prácticamente imposible debido a la complejidad que representa modelar cualquiera de ellos. Por tanto, las técnicas de filtrado usadas son aproximaciones que intentan minimizar las componentes ruidosas en la señal a tratar.

Algunas de las técnicas más comunes usadas en las competiciones de P300 son:

Frecuencias

La primera aproximación para intentar minimizar el ruido de la señal, es el uso de filtros en frecuencia. En primer lugar, se eliminan las componentes superiores a 80 Hz [12], debido a que la actividad cerebral no posee componentes de mayor frecuencia que impliquen procesos cognitivos. Este rango de frecuencias comprende a movimientos musculares, o distorsiones y componentes externas al registro de la señal. Otra componente a eliminar es la frecuencia de la red eléctrica, que puede estar a 50 o 60Hz (dependiendo de la red del país).

Una vez eliminadas las componentes estándar de adquisición de EEG, se localiza el espectro en frecuencia al que la señal de control pertenece. En el caso de P300, dado que la amplitud oscila entorno a los 200ms y el tiempo aproximado necesario para manifestarla es de 300ms, la señal se encuentra entorno a los 5 y 3 Hz [3] [2]. El paso banda usado normalmente en competiciones de P300 es entorno a 0 y 30 Hz. El objetivo es el de minimizar el ruido pero sin perjudicar las componentes deseadas en la construcción del filtro. El filtro entorno a 0, es importante debido a que algunas de las métricas de medición de P300 toman como referencia la amplitud relativa entre periodos de la señal. Y dado que el periodo estándar de estudio son 600ms, aquellas componentes de muy baja frecuencia impiden hacer este tipo de análisis.

Remuestreo de la señal

Aunque no es una técnica usual en el análisis de EEG, el remuestreo de señal es interesante de cara a la clasificación de la señal. La razón para remuestrear esta, se basa en el filtro previamente realizado y la necesidad de minimizar las componentes del vector a clasificar [10]. Es decir, dado un filtro que elimine aquellas componentes superiores a una frecuencia f , donde f es el límite superior del paso banda, solo es necesario muestrear la señal a $2f$ Hz (frecuencia de Niquist). En el caso de EEG, y un filtro entre 0.5 y 30 Hz, el número mínimo de muestras por segundo es 60. De este modo, la señal puede ser remuestreada a 60Hz, independientemente de la frecuencia de muestreo original. El cálculo de los puntos de la nueva señal remuestreada, es realizado mediante el promedio de cada conjunto de n puntos en uno nuevo. Donde n equivale al factor de reducción, o de remuestreo. En términos prácticos, los n primeros valores producirán un único valor resultante, y al realizar el proceso sobre el total, la señal reducirá su frecuencia de muestreo a $\frac{1}{n}$.

Reconstrucción de la señal por estímulos

Como ya se ha comentado en 2.1.2, los periodos de evento tras un estímulo en P300, son superiores al intervalo entre los estímulos. Por tanto, es necesario reconstruir la señal en función de los estímulos. Esto es necesario debido al propósito de P300 que es detectar, en promedio, el estímulo tras el cual se produce un aumento en la señal. De este modo, se establece un vector para cada estímulo mostrado, que contiene las t muestras siguientes a la aparición del estímulo. El periodo estándar a capturar suele ser 600ms, aunque en algunas soluciones BCI se han cogido intervalos posteriores a 0 para evitar capturar el P300 producido por un estímulo previo.

En este proyecto, se ha seguido la estructura mostrada en la figura 2.10. En ella, se puede observar como la época de caracter se almacena en forma de matriz, cuyos ejes representan el estímulo y el número de aparición en la época. Del mismo modo, cada casilla contiene el vector de muestras registrado tras la presentación del estímulo. Este proceso, no obstante, solo engloba un canal y por tanto debe de ser producido independientemente para cada electrodo fuente.

2.2.2. Extracción de Información: Aplicaciones comunes en EEG

Tras tratar la señal para minimizar la cantidad de ruido, es interesante estudiar las características definitorias del sistema. Entendiendo estas como la información mínima necesaria para representar la serie. Reducir la dimensionalidad o reinterpretar las fuentes de datos, son algunos ejemplos interesantes que permiten analizar la información de forma óptima. En este caso, mediante la identificación parcial del ruido existente o la localización de parte de la información relevante. No obstante, es posible estudiar distintas características de los datos relacionadas con su información. La estructura, complejidad o componentes marginales, son algunos ejemplos interesantes de cara a maximizar su separabilidad.

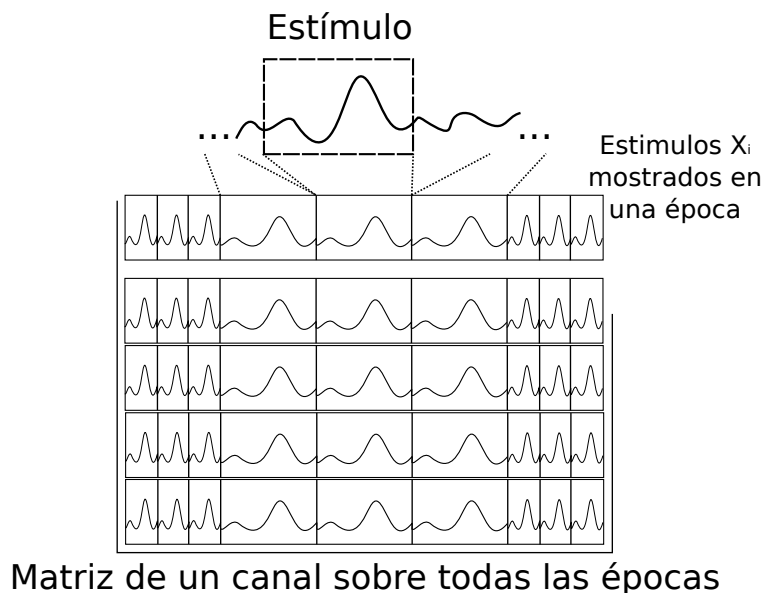


Figura 2.10: Esquema de representación de la reconstrucción de la señal por estímulos usado en el proyecto

Por otro lado, el desarrollo de un sistema de clasificación, permite separar las distintas clases de eventos. Esta componente es fundamental, debido a que proporciona la capacidad de discriminar necesaria para el funcionamiento del sistema BCI en conjunto. Y mediante diferentes métodos, permite establecer de forma automática, dado un conjunto de datos, aquella o aquellas fronteras de separación entre clases. En contra partida, la capacidad de discriminación de estos sistemas suele estar limitada al estricto estudio de separación de clases. Esto es, un estudio supervisado sin conocimiento alguno de la naturaleza de los datos [19]. Aunque existen otras limitaciones como el tiempo de respuesta o la capacidad computacional.

En el caso del desarrollo de sistemas BCI, como ya se ha comentado, los principales problemas son: no estacionaridad de la señal, incluyendo no linealidad, ruido y deriva de la propia respuesta a estímulos; gran cantidad de fuentes de datos, número de electrodos sin conocimiento previo de la calidad de su información; variabilidad entre sujetos, y por ello no generalizable; y en el caso de P300, difícilmente separable por un clasificador normal en la mayoría de los casos. Algunas soluciones BCI proponen el uso de metodologías adaptativas para intentar abarcar este tipo de problemas [24]. Generalmente, para abordar algunos de estos problemas se utilizan técnicas de análisis y tratamiento de información como: reducción de dimensionalidad, estudio del ruido en frecuencia y tiempo, o extracción de componentes [14].

Debido a que la cantidad de técnicas de información es grande y variada, en esta sección solo se exponen algunos de los ejemplos más usados en problemas con datos de EEG, relacionados con P300.

Análisis de componentes principales (PCA)

La cantidad de electrodos disponibles para la toma de datos en EEG, normalmente, alcanza los 64 electrodos útiles. En la mayoría de los paradigmas de EEG, se definen referencias de electrodos en función de la señal de control [2] [10]. No obstante, estas referencias se establecen en promedios de eficacia. Como ya se adelantaba en la sección 2.2.2 la no estacionaridad y variabilidad de un sistema vivo, hace bastante complicado sacar valores generales para todos los sujetos. Tanto es así, que en algunas soluciones BCI [10] se proponen algoritmos de búsqueda

de electrodos óptimos para la persona. En la mayoría de los casos suponen algoritmos costosos, o incluso muy específicos a la situación en particular.

Una solución aproximada para este problema, es la aplicación de sistemas de reducción de dimensionalidad. Un ejemplo es PCA, que permite reducir el número de fuentes de datos, en este caso, de 64 canales a N ($N < 64$). Para ello se construye la matriz de covarianza de las distintas señales, y se crean nuevas dimensiones a través de los autovectores. Del mismo modo, haciendo uso de los autovalores, se puede valorar la expresión de la información que el autovector correspondiente expresa de la matriz original. En el caso de PCA, dado que la matriz es de covarianza, la "información" valorada es la varianza. Por tanto, se pasa de un conjunto con K dimensiones a otro con el mismo número, pero con la posibilidad de seleccionar aquellas que expresen la mayor parte de la varianza. En el caso anterior, las N dimensiones finales explicarían un porcentaje de la varianza inicial.

En el caso de EEG, PCA permite reducir de forma automática el número de fuentes. De este modo, las componentes inherentes específicas del problema a solucionar, son resueltas para cada conjunto de datos de forma inmediata. Del mismo modo, es posible aislar parte del ruido de la señal en dimensiones finales con un bajo porcentaje de información.

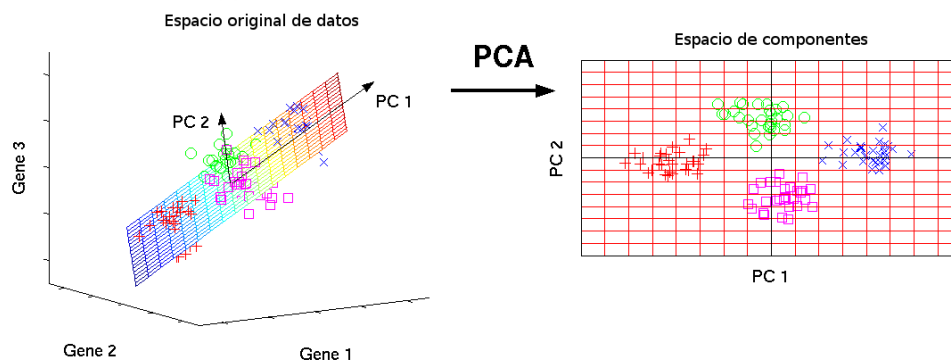


Figura 2.11: Esquema de representación de PCA. Adaptado de: http://phdthesis-bioinformatics-maxplanckinstitute-molecularplantphys.matthias-scholz.de/fig_pca_illu3d.png

Análisis de componentes independientes (ICA)

En contraposición con el caso de PCA, ICA no intenta aislar la información concentrando su varianza en N dimensiones. Por el contrario, intenta separar aquellas componentes no relacionadas. En otras palabras, expresar cada dimensión como una combinación de N fuentes de información e intentar discriminarlas para cada una de las fuentes iniciales [25] [26]. En términos formales, expresar cada variable de entrada (x) como la combinación de las N variables de salida (s) siguiendo el siguiente ejemplo:

$$x_i = a_{i,1}s_1 + a_{i,2}s_2 + \dots + a_{i,k}s_k$$

De este modo, ICA intenta separar componentes estrictamente no relacionadas [9]. Para ello, es necesario primero definir un criterio de relación que nos permita discriminar cuando dos variables están o no relacionadas. Un criterio comúnmente usado es el de Información Mutua (MI). Cuando la MI de dos variables es cero, quiere decir que no comparten información y por tanto no tienen relación. Aunque este es un método estándar, existen otros métodos para discriminar cuando dos variables están o no relacionadas (por ejemplo estadísticamente).

En el caso de EEG, ICA puede resultar útil para detectar diferentes procesos de la actividad cerebral registrados en el conjunto de electrodos [14]. No obstante, debido a la gran cantidad de procesos y la no estacionaridad de estos, el uso de ICA resulta en una búsqueda ciega. Aunque puede resultar útil como medida de reducción de incertidumbre frente al problema o como análisis puntual de las fuentes de la señal.

Transformada Wavelet discreta (DWT)

Un problema de los filtros basados en análisis en frecuencia es la posibilidad de no identificar componentes no estacionarias. Esto es, componentes en frecuencia que no se manifiesten durante la totalidad de la serie. De este modo es más preciso el uso de técnicas de estudios de componentes en frecuencia y en tiempo. Es el caso de la transformada wavelet discreta. Al contrario que la transformada de Fourier, la transformada wavelet mide las componentes en frecuencia con lo que se conoce como "wavelet madre". A efectos prácticos, en lugar de medir las componentes en frecuencia con funciones infinitas (senos y cosenos periódicos), hace uso de una finita, la "wavelet madre". Esta se rescala y transforma para captar las distintas componentes en frecuencia. A mayor frecuencia menor longitud de la serie, lo que causa que las componentes de mayor frecuencia tengan mayor precisión en tiempo. Por tanto las componentes de menor frecuencia son menos precisas de identificar en tiempo.

En el caso de EEG, el uso de DWT es interesante debido a la no estacionaridad de la señal inherente a un sistema vivo [14]. Esto se traduce en fenómenos no permanentes en la señal que pueden contaminarla en gran medida. En algunas soluciones BCI se ha aplicado DWT para localizar estos periodos, y en algunos casos filtrar en este espacio de forma más precisa.

2.2.3. Extracción de Información: Otras técnicas

Aunque las técnicas revisadas conforman un subconjunto bastante representativo de las aplicaciones en sistemas BCI, existen otras técnicas interesantes aplicadas en problemas con EEG. En este apartado se introducen, además de estas, algunas propuestas para medir la información contenida en EEG, y más concretamente en el P300 evocado, de cara a su clasificación.

Complejidad Predictiva

La técnica de complejidad predictiva se basa en el estudio de la historia necesaria para anticipar el futuro de una serie temporal de forma óptima [22] (sustituyendo a [27]). Esto conforma la métrica de complejidad del sistema. Para ello hace uso de cadenas de markov producidas a partir de un algoritmo de generación dada una serie con un pasado máximo [28]. No obstante, para ser rigurosos con la definición inicial, es necesario definir el concepto de "anticipar óptimamente". Para ello, se construye una función de utilidad a maximizar en función del futuro esperado y el producido. Continuando en esta tendencia, se simplifica la cadena de markov estableciendo relaciones de equivalencia entre los estados, en función de esta función de utilidad. A estos conjuntos de estados formados por las relaciones de equivalencia, se los denomina "estados decisionales".

La información mutua entre la predicción producida por los estados decisionales y la serie real determina la complejidad predictiva de la señal (Haciendo uso de la complejidad estadística introducida en [29]). Así es posible diferenciar entre periodos de la señal más regulares, y por tanto más fáciles de predecir, y otros con mayor complejidad y menor predecibilidad. Estas diferencias en la complejidad de la señal permiten detectar cambios en su evolución.

Vectores de Características Multifractal

Al igual que el método de Complejidad Predictiva, este busca capturar la complejidad de la serie [22]. Para ello se define la unidad de medida de complejidad, a través de la dimensión de Hausdorff [30]; y el cálculo de la regularidad de la serie, mediante el exponente de Hölder y el teorema de Taylor. En primer lugar, se descompone la señal $x(t)$ en un polinomio $p(t - t_0)$ y un residuo $r(t - t_0)$, con cierto grado. A continuación se asume que existe una constante c tal que

$$|x(t) - x(t_0)| \leq c||t - t_0||^h$$

El grado h más alto, si existe, para el cual tanto $p(t - t_0)$ y $r(t - t_0)$ se ajustan, se denomina el exponente local de Hölder de x en t_0 . Este permite medir la fuerza de la divergencia del residuo. Para terminar, se estima la complejidad del conjunto de grados h para cada t_0 . Esto se obtiene, mediante el cálculo la dimensión de Hausdorff de la distribución de puntos con mismo exponente h . Y de este modo, se obtiene la distribución de complejidad de los distintos exponentes de Hölder.

Existen diferentes técnicas para el cálculo de h en cada punto de la serie. La más genérica, como se explica en [31], es a través del uso de los cumulantes de la DWT, pero representa un computo lento y por tanto, ineficaz en sistemas BCI. No obstante otras técnicas como la estimación por la transformada de Legendre de los momentos, permiten hacer uso de este tipo de técnicas en tiempo real.

Medidas de similitud

En este proyecto se introduce la idea de extraer características de la señal mediante el estudio directo de las similitudes inherentes a la definición de la señal de control, en este caso P300. Reconstruyendo una señal, como se explicaba en 2.2.1, como un conjunto de intervalos de tiempo por estímulo, es posible aplicar distintas técnicas de similitud o distancia (métrica directamente opuesta a la similitud) para intentar relacionar aquellos periodos con características comunes. Para ello es necesario definir una reconstrucción de los datos, por intervalo, que garantice maximizar las propiedades de los conjuntos que se busca relacionar. La razón de reconstruirlos por intervalos, es para verificar la no supervisión del proceso. Un proceso con conocimiento de clase previo, y en un caso como el de P300 con clases tan desbalanceadas, puede llevar a conclusiones erróneas.

En el caso de P300 en EEG, como ya se adelantaba en 2.1.2, su comportamiento solo es detectable en promedios debido a: la variabilidad de la adquisición de datos [3], EEG; la no estacionaridad del sistema, con la apariencia de procesos paralelos; la ausencia de un modelo preciso de comportamiento constante, en términos de amplitud y duración; y la manifestación no garantizada tras cada estímulo positivo. Como solución a estos problemas, se propone definir un intervalo de P300, como promedio de varios intervalos de P300. Estos intervalos se denominarán "series" y "ventanas", respectivamente, para discriminar mejor entre la reconstrucción de los datos, y la fuente original.

La cantidad de medias usadas en una serie de P300, sin embargo, no puede ser ilimitada debido a que la propia definición de la señal establece una no estacionaridad. Por ello es necesario establecer unas cotas a la cantidad de ventanas de P300 usadas en cada serie. Para abordar este problema, se realizan series distintas con ventanas diferentes. No obstante, para no producir cambios bruscos en la estructura de la señal de P300, se determinan conjuntos solapados de ventanas para cada serie. El objetivo de este tipo de técnicas en este problema será el de captar la estructura local del P300, o al menos, facilitar la capacidad de discriminarlo.

Correlación La correlación entre dos series se define como el la relación bidireccional, en términos de comportamiento, entre ambas. No obstante existen varias formas de medir la similitud en el tiempo de dos señales. En este proyecto se ha hecho uso de la correlación cruzada, que hace uso del producto desfasado entre las series; y la correlación estadística, que mide el factor entre la covarianza de ambas series y el producto de sus varianzas. En el caso de la correlación cruzada, se obtiene una serie no normalizada, y se retorna el valor máximo obtenido. Mientras que en el caso de la correlación estadística, se obtiene un valor normalizado y único por prueba.

Estas dos técnicas miden la relación que existe entre dos series dadas, con la diferencia de que la correlación cruzada nos permite obtener el desfase entre ambas y un valor no normalizado de su similaridad. Con lo que podemos concluir que la primera proporciona una resolución en fase, y la segunda en únicamente en comportamiento.

Información Mutua (MI) La MI se define como la cantidad de información que comparten dos variables aleatorias. Es decir, la cantidad de información compartida por dos sucesos observables, pudiendo definirse estos para conjuntos de datos. Para poder hacer uso de esta métrica es necesario definir la medida de información, en este caso la entropía de Shannon. Una vez definida la medida de información base, solo es necesario definir la MI como la suma de la cantidad de información de cada variable menos la unión de ambas. Donde $MI(x, y)$ equivale a la MI entre x e y , y $H(x)$ a la cantidad de información de x , podemos expresar MI como:

$$MI(x, y) = H(x) + H(y) - H(x|y)$$

La entropía de Shannon, de forma breve, define la cantidad de "bits" necesarios para representar la información contenida. Del mismo modo, Shannon formaliza la medida de bit como unidad de información mínima, representada como la información máxima de una variable aleatoria de dos valores. Para ello, la variable debe ser equiprobable respecto a sus dos posibles valores. La definición formal se representa como:

$$H(x) = \sum_{i=1}^N p(x) \log \frac{1}{p(x)}$$

Para poder hacer uso de esta MI como unidad de similaridad, es necesario expresarla normalizada de la siguiente forma:

$$NMI(x, y) = \frac{MI(x, y)}{\sqrt{H(x)H(y)}}$$

Distancia por compresor normalizada (NCD) Al igual que MI, NCD tiene el objetivo de medir las diferencias en la información contenida de, en este caso exclusivamente, dos fuentes distintas. La diferencia es que MI producía un coeficiente en base a Shannon midiendo la fuente de manera directa. En el caso de NCD, es necesario construir un objeto, fichero de datos, para definir su cantidad de información relativa a si mismo mediante un compresor. Es decir, la propia composición del fichero definirá su capacidad de reducción tras aplicar un compresor.

Un compresor, en términos de información, es un algoritmo que intenta, mediante el estudio de la estructura del objeto, localizar la representación óptima para este [32]. Es decir, descubre la codificación que minimiza el tamaño de la representación final del mismo. En función de la combinación de compresor y estructura del objeto, la nueva codificación del fichero será más o menos óptima. Este suceso de compresor se basa en que cada compresor intenta identificar la estructura del fichero de forma distinta.

Para terminar, las medidas de compresión, al igual que MI, no producen directamente una salida de similaridad comparable. Por el contrario es necesario establecer una métrica que compare los nuevos objetos. Del mismo modo que con NMI, se establece una métrica que compare la información de los objetos en conjunto y por separado. En este caso, mediante concatenar los dos objetos, de modo que en caso de que sean iguales no se vea prácticamente alterado.

$$NCD(x, y) = \frac{Z(xy) - \min\{Z(y), Z(x)\}}{\max\{Z(y), Z(x)\}}$$

Donde $Z(x)$ representa el compresor aplicado sobre el objeto x , y $Z(xy)$ aplicado sobre la concatenación de ambos. De este modo definimos una unidad de similaridad opuesta. Es decir, medir la similaridad mediante la distancia normalizada de ambos objetos, tras su compresión. En el caso de EEG, es necesario describir una estructura de datos a comprimir. Como punto de partida, en este proyecto se propone una estructura simple para intentar maximizar las posibilidades de encontrar un compresor óptimo, y por tanto, de detectar la estructura del P300 al compresor [33].

Synchronization Likelihood (SL) Continuando con la premisa de que la generación del P300 es dependiente de la fuente de datos, y no se manifiesta siempre igual, es interesante hacer uso de técnicas que comparen distintas fuentes de datos pero que permitan identificar sus diferencias [34]. Este es el caso de SL, que se define como la relación, dado un umbral, que existe entre la sincronización de una señal consigo misma y la de otra [35]. Para medir esto, se establece un umbral y se construye un vector para ciertos puntos de la serie en base a unos parámetros establecidos para identificar ciertas componentes en frecuencia. De este modo, para cada punto x_i de la serie se define como un vector de modo que:

$$x_i = (x_i, x_{i+L}, x_{i+2L}, \dots, x_{i+(m-1)L})$$

En este caso, L y m se establecen una función de las frecuencias de muestreo y corte. Esto es necesario para establecer un máximo de puntos a muestrear, de modo que no se complique más tarde la medida de distancias por el ruido de muestreo; y un mínimo, para no perder las componentes en frecuencias deseadas. La definición de cada punto como un vector nos permite, mediante la norma, medir la distancia entre cada instante de la serie. Y así, registrar aquellos momentos en los que se considera que la señal se ha sincronizado, bajo un umbral establecido previamente.

Finalmente, para poder establecer la métrica de comparación, se establece una función escalar θ para la distancia, y se realiza el producto a lo largo de la señal promediado.

$$S_i = \frac{1}{N} \sum_j \theta(\epsilon_y - |y_i - y_j|) \theta(\epsilon_x - |x_i - x_j|)$$

No obstante, SL causa problemas a la hora de comparar diferentes fuentes de datos. La razón es que si una de las series a comparar posee una regularidad grande, o en términos de SL, se sincroniza muy a menudo consigo misma, siempre tendrá más posibilidades de sincronizarse con cualquier otra. Hecho independiente de si realmente existe una relación entre ellas. Para evitar este tipo de problemas, se pueden establecer ϵ distintas para cada serie, lo cual lleva un incremento en el coste computacional debido a la búsqueda, o intentar normalizar los coeficientes entre distintas comparaciones.

Representación de la estructura de datos: Maketree

Los métodos de similitud revisados en la sección previa (2.2.3), proporcionan un coeficiente de la cantidad de información compartida, para distintas fuentes de entrada. Y aunque cabría esperar que este permitiese formar subconjuntos con aquellas fuentes más parecidas (Por ejemplo, uniendo aquellos conjuntos cuyo parecido entre todos los pares sea superior a un valor), en algunos casos la distribución de valores no es separable mediante umbrales absolutos. En algunos casos ni siquiera se distribuye uniformemente, lo que dificulta en gran medida la separación directa entre grupos o clusters. De hecho, para algunas técnicas (por ejemplo, NCD 2.2.3) los valores se concentran entorno a un intervalo ínfimo, y resulta complicado establecer las fronteras para formar subconjuntos.

Para evitar estos problemas, en este proyecto se ha optado por construir un árbol binario mediante el algoritmo propuesto en [36]. Este método define un árbol de nodos, en función de las quarter topologies, de los subgrupos de estos. Esta topología se define como dentro de un conjunto de nodos, cuatro nodos hojas, cuyos pares de caminos no se crucen. Y el árbol objetivo, en este caso, es aquel que minimiza la distancia entre las parejas de nodos que cumplen esta propiedad. No obstante, escoger el subconjunto de ramas que minimicen la distancia del grafo resulta un problema de orden NP (nondeterministic polynomial time). Y en el caso de este proyecto, aún más debido a que se trata de un grafo pesado y normalmente completo. Por lo que para el cálculo de este se hace uso de heurísticas para acelerar su proceso. En este proyecto se ha hecho uso de la herramienta *maketree*, incluida en el paquete *Complearn*.

Esta herramienta posee limitaciones, debido a su implementación y a la naturaleza del problema. En primer lugar, su heurística se basa en una semilla aleatoria que itera intentando encontrar una convergencia en la calidad del grafo. Esta se mide por la unidad de minimización propuesta en [37]. Por ello, ciertas semillas iniciales o fallos a la hora de adelantar la convergencia del proceso, pueden hacer este muy lento e incluso, a efectos prácticos, interminable. Por otro lado, la capacidad de la heurística de la herramienta conjunto con la capacidad hardware disponible, impide superar cierto número de nodos a la hora de computar el grafo. Limitaciones muy problemáticas de cara a la simulación de parámetros, pero con soluciones razonables que se explicarán en profundidad en las secciones 3.3.2 y 4.2.

2.3. Sistemas de clasificación

El ejercicio final, y más importante, de un sistema BCI es la clasificación de los eventos de cada época. El filtrado de la señal y la extracción de información de la señal, previamente introducidas en las secciones 2.2.2 y 2.2.1 respectivamente, facilitan la tarea al clasificador pero no producen, normalmente, resultados por si solas. Aunque en algunos casos, la aplicación de técnicas de información es suficiente y solo es necesario de un clasificador simple para establecer la frontera de decisión (es el caso de un umbral). De cualquier modo, el clasificador define el componente final del proceso BCI dando la capacidad de discriminar que estímulos, en el caso de señales de control evocadas, producen respuestas en el sujeto y cuales carecen de ellas. En esta sección se revisarán algunos de los clasificadores más usados en soluciones BCI:

2.3.1. Análisis de Discriminante Lineal (LDA)

Al igual que PCA, LDA establece un criterio para reducir la dimensionalidad de un conjunto de datos [14]. En este caso, en lugar de intentar concentrar la varianza en dos nuevas dimensiones, el objetivo es maximizar la distancia entre los vectores de cada clase. Por tanto, se trata de un método supervisado de reducción de dimensionalidad. El criterio general para medir la

separación de las clases, es maximizando la distancia entre las medias de cada clase y minimizando la dispersión de cada una por separado. Aunque este método no es en si un clasificador, si que proporciona la capacidad para funcionar como tal. La razón es que al intentar separar los conjuntos de datos de dos clases en una dimensión, conceptualmente hablando, establece una frontera.

El uso de LDA como clasificador es relativamente simple, y proporciona una relación potencia-rendimiento muy aceptable. Esto se debe a la simpleza del método de cálculo de la frontera. Como consecuencia de esto, este método es usado con frecuencia en la clasificación de datos en el contexto de BCI [16] [38] [19]. No obstante, justo por esta sencillez, LDA representa un método con debilidades. Debido a que es un método que asume unimodalidad gaussiana en la distribución de los puntos, es fragil ante conjuntos de puntos con otro tipo de distribuciones. Además al tratarse de una transformación lineal, no permite ajustarse a ciertos tipos de problemas (Por ejemplo, con medias superpuestas). No obstante existen variantes de LDA, como QDA (Análisis de Discriminante Cuadrático), que permiten ajustarse mejor a estos problemas.

2.3.2. Máquina de Soporte Vectorial (SVM)

Una SVM es un clasificador que hace uso de un hiperplano o un conjunto de varios hiperplanos, para establecer fronteras de separación entre vectores de características de varias clases. En este caso, para establecer las fronteras de separación se hace uso de los "Vectores de soporte", que conforman el "Margen" de la frontera. La utilidad del margen es la de conformar un conjunto de vectores de soporte para establecer la frontera de decisión. A un mayor margen, mayor será el conjunto de vectores y mayor la resolución de la frontera. Del mismo modo, la complejidad del problema aumentará, y por tanto, el coste computacional. Una frontera con mucha definición hace al sistema obtener artefactos producidas por el exceso de muestreo. Y por tanto, errores en la clasificación producidos por el sobreajuste del sistema.

Para solucionar estos problemas, se establece el concepto de "Soft margin". Este define un área entorno a la frontera, abierta a fallos en la clasificación. De este modo, permite una generalidad de cara a nuevos datos, menor coste computacional, y la capacidad para ajustarlo de manera externa. En este mismo sentido, se establece el uso de kernels para mejorar la definición de las fronteras de decisión, con un coste computacional razonable.

En el caso de EEG, las SVMs son comúnmente usadas [19] [14] [10], por la relación entre definición y coste de explotación. Aunque construir en clasificador a partir de un conjunto de entrenamiento puede conllevar un coste considerable, y más teniendo en cuenta la búsqueda paramétrica usual, el coste de la explotación del mismo es mínimo. Teniendo en cuenta la necesidad imperiosa de tiempo real usual en los sistemas BCI, SVM resulta una elección de clasificador razonable para casi cualquier sistema BCI.

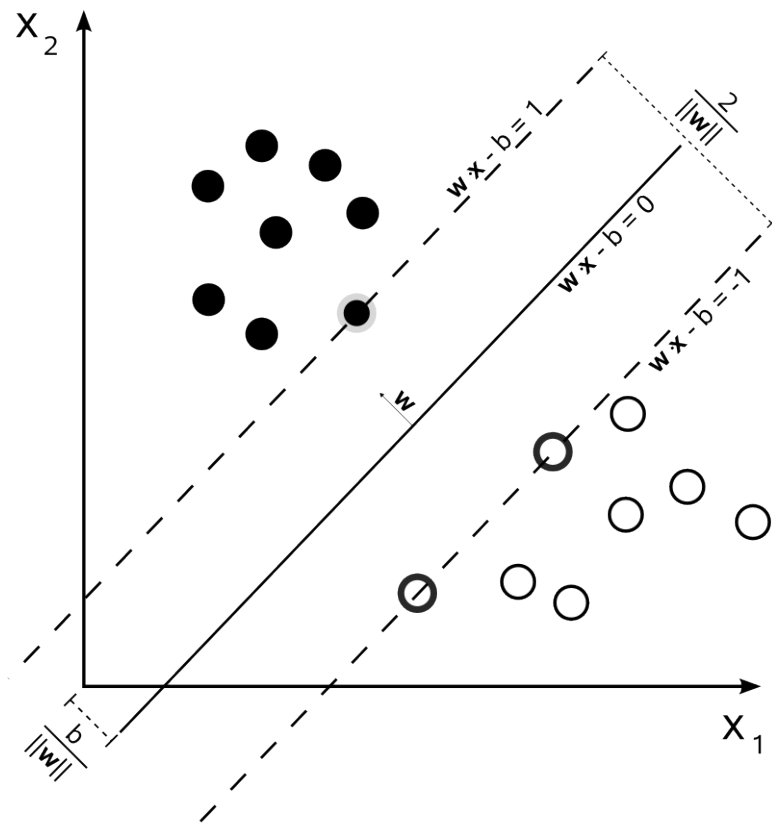


Figura 2.12: Esquema de representación de la frontera de decisión en una SVM. Adaptado de:
https://upload.wikimedia.org/wikipedia/commons/2/2a/Svm_max_sep_hyperplane_with_margin.png

3

Metodología

3.1. Entorno de trabajo

Debida a la variabilidad de técnicas a revisar, así como la incertidumbre inherente a los problemas de EEG, en este proyecto se estableció un entorno de trabajo para el desarrollo del mismo. Las principales razones han sido:

- Homogeneizar resultados, desarrollando pruebas que permitan relacionar las distintas técnicas
- Unificar el sistema de desarrollo, para agilizar el trabajo entre distintas funciones
- Orientar el desarrollo de forma modular y no destructiva, haciendo útiles las funciones realizadas para trabajos futuros
- Y mantener una transparencia de datos, tanto de uso como de fuente de procedencia, para más tarde realizar las pruebas con distintos conjuntos de datos e implementar de forma más simple las funciones a aplicar sobre el conjunto.

El entorno de trabajo se ha establecido, en primer lugar, conforme a una fuente de datos inicial. Esta primera aproximación al problema es interesante debido a que al hacer uso de un conjunto de datos conocido y con resultados, se pueden minimizar los problemas surgidos de este. Las soluciones de este conjunto de datos como referencia en el desarrollo o la literatura existente entorno al mismo, son algunos ejemplos de las ventajas de iniciar el proyecto con un conjunto de datos conocido.

3.1.1. Estructura de datos

Para mantener la transparencia de datos, se desarrolló una función que permitiese encapsular una sesión de EEG, en un array de tres dimensiones. En este caso, los parámetros de la estructura representan los estímulos, las apariciones de estos y el tiempo. Del mismo modo, todas las funciones aplicadas sobre la estructura, carecen de especificaciones de tamaño, con lo

que mantiene una transparencia de cara al origen de los datos. Un esquema de esta estructura se muestra en la figura 2.10.

Cada fila de la imagen representa un estímulo de la sesión, mientras que las columnas son agregadas por cada iteración del conjunto de estímulos. Por último, cada casilla representa un array que contiene las t muestras siguientes a la aparición del estímulo. Aunque este puede ser modificado para almacenar un intervalo no obligatoriamente consecutivo a la aparición del estímulo (Por ejemplo 100ms tras 50ms desde la aparición del evento).

Esta estructura posee limitaciones, establecidas por la propia definición del problema. Como se define en la sección 2.1.3, este tipo de paradigma posee diferentes niveles de descripción, y para poder operar desde el nivel más básico, esta estructura solo permite encapsular una sola época de caracter. Por otro lado, dado que la señal obtenida por cada electrodo no es comparable, esta estructura agrupa ventanas de un solo electrodo (como se adelantaba en la sección 2.1.1).

3.1.2. Métodos de análisis usados

En el desarrollo de este proyecto, se han usado diversos métodos de análisis de información. Algunos de ellos fueron usados de librerías públicas, mientras que otros fueron implementados en el propio desarrollo del proyecto. Entre los sistemas usados podemos definirlos en conjuntos de subsistemas de la siguiente forma:

1. Manejo de datos

- a) **Filtrado de la señal** (*Matlab*): Desarrollado en el proyecto como librería independiente y descrito en la sección 2.2.1, permite eliminar parte del ruido de la señal. Adicionalmente permite reducir el número de datos si fuera necesario.
- b) **Conversión a la estructura de datos** (*Matlab*): Implementación básica al inicio del proyecto para manejar los datos de forma transparente, como se explica en 3.1.1.
- c) **Construcción de objeto o fichero** (*Matlab*): Operación básica desarrollada durante el transcurso de las simulaciones, para la comparación mediante técnicas de compresores. Explicado en profundidad en la sección 3.1.3 y 3.3.1.

2. Métodos de similitud

- a) **Máxima correlación cruzada** (*Matlab*): Modificación de la librería de correlación cruzada de *Matlab* para el proyecto. Explicado en profundidad en 3.1.3 y 2.2.3.
- b) **Correlación estadística** (*Matlab*): Implementación de la definición clásica de correlación estadística para la comparación de fuentes de datos. Explicado en profundidad en 3.1.3 y 2.2.3.
- c) **Synchronization Likelihood** (*Matlab*): Implementación simple del algoritmo de relación de fuentes descrito en [35]. Descrito en las secciones 2.2.3 y 3.2.1.
- d) **NCD (bzlib y zlib)** (*Complearn*): Librería externa usada para el cálculo de distancias mediante compresores. Definido en la sección 3.2.2.

3. Simulaciones

- a) **Cálculo de los coeficientes de Silhouette** (*Python*): Desarrollado internamente en el grupo de investigación, y usado como métrica para el cálculo de calidad de clusters. La base del método se define en 3.1.3 y 3.3.2.

- b) **Iteración automática de los parámetros de fichero** (*Python & Bash*): Implementando para agilizar la obtención de resultados. Conformando la iteración, construcción y organización de los ficheros, y el conjunto del proceso ya mencionado de similitud y calidad de clusters. Proceso descrito en 3.3.
- c) **Búsqueda automática de los sensores óptimos** (*Python & Bash*): Proceso implementado como aproximación, para la ponderación de distintas fuentes de datos. Explicado en profundidad en 4.2.4.
- d) **Construcción y conversión del grafo** (*Complearn & Python*): Conjuntamente con el uso de librerías externas y el desarrollo de sistemas de conversión de la salida de los datos al sistema, se crean y procesan grafos para ajustar la representación a la matriz de distancia inicial. Definido en 3.3.2.

4. Clasificadores

- a) **Clasificador simple** (*Matlab*): Sistema desarrollado durante el proyecto, haciendo uso de las librerías de aprendizaje de (*Matlab*), y el sistema de filtro ya comentado. Este se explica en la sección 3.1.4 de manera más completa.

3.1.3. Sistemas de similaridad

Varios de los métodos revisados en la sección 2.2.2, permiten obtener una medida de proximidad o similaridad entre dos elementos. El caso de las medidas de distancia es una métrica opuesta a la similaridad, pero suficiente a efectos prácticos. No obstante, no todas las medidas que se han revisado devuelven un retorno normalizado, lo cual por tanto, requiere de un proceso extra para su comparación.

Para comparar estos métodos, se estableció una metodología de similaridad o distancia, única para todas las métricas. Y de ella, un sistema transparente al método usado para poder obtener resultados comparables. El sistema se desarrolló en *Matlab* (con la excepción de la comparación final mediante NCD), para facilitar la operabilidad con código proveniente de otras competiciones BCI, y por ser más usado en general en este tipo de problemas. No obstante, para mejorar el control automático de las simulaciones también se hizo uso de código en *Python* y en *Bash*, explicado en la sección 3.3.

Formato de datos

Debido a la naturaleza del problema, y con el objetivo de establecer una comparación razonable de distintas fuentes de datos, es necesario reconstruir estas como se adelantaba en la sección 2.2.3. Los nuevos conjuntos de datos serán, por tanto, conjuntos de series concatenadas. Siendo cada serie el producto de promediar varias épocas de P300 (ventanas). En la figura 3.1, se muestra de forma simplificada el proceso de construcción.

En primer lugar se extrae la ventana asociada a cada estímulo de la señal. Este proceso está integrado dentro de la creación de la estructura de datos interna ya definida en la sección 3.1.1. De modo, que una época de carácter produce una matriz de tipo de estímulo por aparición en el tiempo. En el ejemplo de la matriz de delecteo usado en este proyecto, se traduce en una matriz de doce estímulos (seis filas y seis columnas), por quince intensificaciones (estimulación de la fila / columna) por cada estímulo. La longitud de la ventana de tiempo, varía en función de la configuración del proceso.

A continuación, por cada tipo de estímulo, se crean promedios de varias ventanas, dado un solape. En este proyecto el solape se ha mantenido constante, con lo que todas las ventanas

compartirán siempre su totalidad de ventanas menos una (el desplazamiento). En la figura 3.1, se representa con el nombre de "Serie" (Promedio de ventanas), que es la denominación asignada en este proyecto al promedio de ventanas. Y para terminar, se concatenan series hasta el mínimo entre el valor especificado y la cantidad de series (para mantener la modularidad del conjunto). En la figura, se muestra como "concatenación de series", como la salida final del proceso.

Tanto la cantidad de series a concatenar como el número de ventanas a promediar, son parámetros que variarán el efecto de las diferentes medidas de similitud. Cabe destacar, que a cada ventana se le añade una marca de tiempo, para el caso de NCD, para facilitar la capacidad de extracción de información al compresor. Del mismo modo que para el caso de los métodos basados en compresores, los datos sufren una conversión de su representación numérica a tipo cadena de caracteres. Esto es importante, debido a que el mismo sistema constituido por archivos binarios con una representación numérica, tipo coma flotante, puede variar los resultados obtenidos.

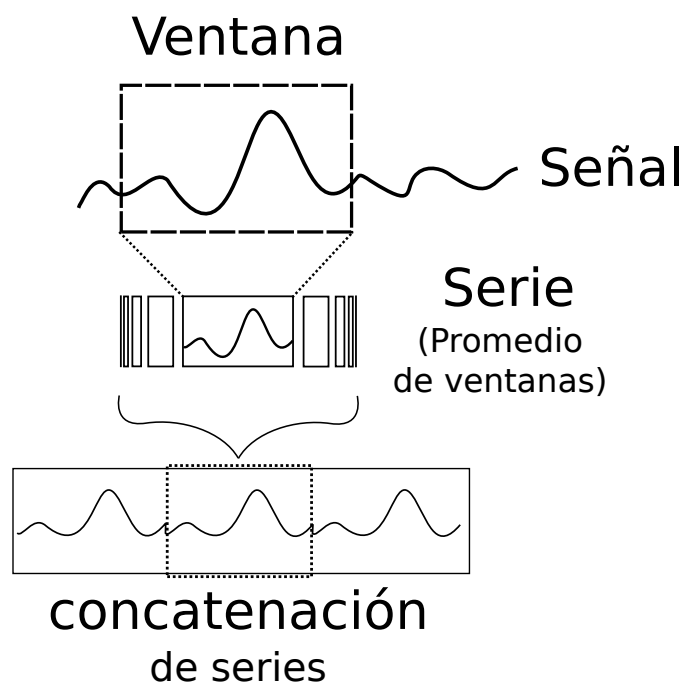


Figura 3.1: Esquema de la construcción de los datos. La señal se procesa como en la figura 2.10, para obtener las ventanas asociadas a evento o estímulo. Cada conjunto de ventanas produce una serie, y finalmente esta se concatena a otras series produciendo la fuente final a comparar.

Un factor importante a tener en cuenta, es que cada época de carácter solo posee un determinado número de ventanas. Por tanto, a mayor número de ventanas por serie, menos concatenaciones de series serán posibles. En la construcción de este formato de datos, así como en las pruebas, se ha asumido el criterio de medias primero. Por lo que en caso de no haber suficientes datos para la construcción total de la nueva estructura, el sistema primero se promediará, y luego se concatenarán todas las series posibles.

Los nuevos subconjuntos de datos asociados a evento, por tanto, vienen definidos por ser una sucesión de series concatenadas, siendo estas a su vez un producto de promedio de ventanas. Perteneciendo cada subconjunto a un estímulo predeterminado. A configuraciones de bajo número de ventanas y series, la cantidad de subconjuntos por estímulo, se incrementa, dando lugar así a diferentes subconjuntos por estímulo. Por último, y con el objetivo de producir una muestra de datos más uniforme, se establece un solape entre las ventanas que conforman cada serie (en caso de que el número de medias sea superior a uno).

Sistema de Representación

Como resultado de aplicar cualquier técnica de similaridad de las revisadas en la sección 2.2.3, se produce una matriz de semejanza (o distancias en el caso de compresores). Esta matriz define la relación establecida por el método usado de cada fuente con todas las demás. Las fuentes a comparar, es decir los ejes de la matriz, se conforman con el formato descrito en la sección previa 3.1.3. En función de la configuración de medias y concatenaciones se conformarán un determinado número de ejes, dado que a mayor concentración de la información por iteración, menos información resulta para el resto de configuraciones. A modo de ejemplo, con doce intervalos asociados a evento, es posible producir seis objetos concatenando de dos en dos, o dos concatenando de seis en seis. Esto se define en profundidad en la sección 3.3.1. No obstante, como se describe en 3.3.2, la cantidad de fuentes de entrada se limitará a una cantidad fija, para eliminar parte de la dependencia de la construcción de ficheros.

Una vez obtenida la matriz de distancias, es necesario hacer uso de métodos adicionales para poder interpretar los datos obtenidos. Esto se debe a que algunos métodos, como se introduce en 2.2.3, el rango de valores de similitud se concentra en intervalos limitados. Por ello, para interpretar los datos se hizo uso de mapas de color y dendogramas. Calculando estos últimos a través de el método definido en la sección 2.2.3.

El sistema con mapas de color, permite ganar intuición de forma relativa entre las fuentes usadas. Es decir, estudiando localmente la variación de similaridad dentro de una misma prueba. Por ejemplo, intentando detectar a simple vista aquellos conjuntos que sean más similares entre ellos que entre los demás. O por otro lado, ver si ciertos conjuntos que comparten características manifiestan un coeficiente alto. No obstante, la ausencia de uniformidad ya comentada, y complejidad de los datos, hizo difícil obtener información a través de este tipo de representación, durante las pruebas desarrolladas a lo largo del proyecto.

En el caso de los dendogramas, dado su definición de árbol, el análisis de información puede hacerse del mismo modo que en un grafo. Como primera aproximación se intento medir la organización de los nodos en forma de subconjuntos próximos, o clusters. Con la asunción de que aquellos objetos no pertenecientes a la clase de evento asociado pertenecen a una clase común, aunque teóricamente carecen de información compartida.

En este tipo de representación se obtiene una idea de distancia distinta a la usada en las matrices anteriores. En este caso, la distancia se define como la cantidad de saltos entre nodos en el árbol generado. Para facilitar el análisis desarrollado en este proyecto se asoció a los grafos generados un color por clase, como se muestra en la figura 3.2. En ella se puede encontrar dos tipos de nodos: Nodos nexos y Hojas. Estos últimos representan los objetos a comparar en el desarrollo de las distintas pruebas. Dentro de estos podemos definir una representación definida de la siguiente forma:

1. **N/Y**, para expresar si pertenecen a la clase de evento asociado "Y" o no "N"
2. **R**, seguido de un número (como los siguientes atributos) indica el subconjunto o palabra a la que referencia (Nivel 2)
3. **SW**, referente a la época de carácter (Nivel 3)
4. **SE**, indicando el estímulo en este caso. Este último solo es añadido como ayuda a la hora de tratar con los datos directamente.
5. **T**, este último no siempre se añade y representa el número de objeto generado. Esto es importante debido a que, como ya se ha comentado, dentro del conjunto de un mismo estímulo pueden generarse varios subconjuntos de series distintos, y por tanto, objetos

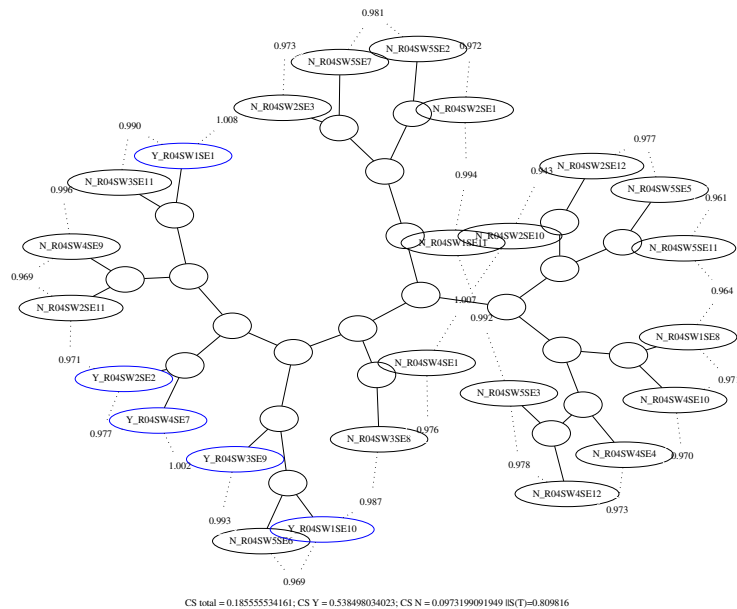


Figura 3.2: Ejemplo del sistema de representación, mediante grafo. Se incluyen los coeficientes de calidad de cluster totales y parciales. Los nodos hoja coloreados en azul, representan a la clase asociada al evento P300.

diferentes. No obstante a grandes configuraciones la cantidad de objetos por estímulo se reduce a uno.

Aunque la representación visual resulta muy potente en muchos de los grafos generados como toma de intuición, la necesidad de automatización, así como de formalización, necesita de una métrica para poder determinar la calidad de un grafo. Por ello, se hace uso de las distancias entre los nodos de cada cluster, como medida de la calidad de cluster. Entendiendo esta como la cercanía de los nodos de cada cluster con respecto al resto de clusters. El Coeficiente de Silhouette permite medir, de forma simple, esto como promedio de la calidad individual de cada nodo con el resto de clusters y el suyo propio. Esta se define en profundidad en la siguiente sección 3.1.3.

Coeficiente de Silhouette

El coeficiente de Silhouette permite medir, de forma simple, la calidad de un cluster con respecto a su vecino más próximo [39]. De este modo, es posible asignar a cada conjunto de nodos, previamente establecido, una métrica de calidad de cluster. Para calcular este coeficiente se define previamente la distancia media de un nodo a un cluster. Esta se establece como la distancia media a cada otro nodo de ese cluster. Esto es doblemente útil, dado que permite medir como de "ceranos", son los nodos de un cluster, y como cada uno de ellos lo está del cluster más próximo. Estableciendo así una métrica de calidad entre clusters, relativa entre ellos, y por tanto normalizada.

En resumen, dada la función de distancia media $f(x_i, C)$, siendo x_i un nodo de un cluster y C un cluster, definimos el coeficiente de Silhouette como la diferencia de la media de la aplicación de esta función, dado un conjunto de nodos, sobre su cluster y el más próximo, dividido entre el máximo de las medias. Es decir, siendo X el cluster del conjunto de nodos x_i e Y , el cluster más próximo:

$$s(X) = \frac{\sum_i^N f(x_i, X) - f(x_i, Y)}{\max(\sum_i^N f(x_i, X), \sum_i^N f(x_i, Y))}$$

Esta métrica también permite medir la calidad total de los clusters de un grafo iterándola sobre el conjunto de nodos total. Obteniendo así una métrica de como de diferenciados están los clusters en un mismo grafo. Para ello solo es necesario alternar el cluster de referencia en cada función dependiendo del nodo.

3.1.4. Sistema de clasificación

Un problema frecuente en la literatura de BCI, y más concretamente entorno a las señales de control, es la desconexión en algunos casos del sistema desarrollado, con el problema inicial. Al contrario que los clasificadores que proporcionan un índice final medible, otros como los filtros o técnicas sobre la información, producen cambios en el sistema sin producir una métrica comparable. Por ello, es interesante el incluir un Clasificador en el desarrollo de este proyecto, para poder tener un dato cuantitativo del efecto del uso de unas técnicas u otras. Aunque es cierto que no es una relación directa, debido a que un mismo métodos no tiene porque producir el mismo impacto en cualquier clasificador, si que es cierto que permite verificar, en casos de mejora, el impacto del mismo. Para ello en este proyecto se ha realizado un clasificador simple, inspirado en las soluciones de varias competiciones BCI [10] [9] [40] [41]. El clasificador desarrollado hace uso de la metodología de transparencia de datos antes mencionada, como punto de partida para el manejo de datos.

Para poder desarrollar un clasificador, es necesario establecer un vector de características. Este conformará una serie de parámetros que el clasificador usará para establecer los criterios definitorios de cada clase. Dicho de otro modo, establecer formalmente que diferencia a una clase de las demás. La elección del vector de características, resulta importante debido a que el exceso, o la ausencia, de información producirá errores a la hora de establecer los criterios de separación entre clases. Por lo tanto, resulta interesante la correcta elección del vector de características.

En primer lugar se estandariza la señal junto con un paso banda, chebyshev de orden 3, entre 0.5 y 30 Hz. Este filtro elimina gran parte del ruido externo y procesos internos que no son interesantes para el análisis (como se adelanta en la sección 2.2.1). A continuación se remuestrea la señal (a través del cálculo de medias de los puntos entorno al nuevo) a 60 Hz, para reducir la cantidad de variables de cara al clasificador. La elección de 60 Hz como nueva frecuencia de muestreo viene motivad por la necesidad de no perder información. Y dado que la componente en frecuencia más alta no pasa los 30 Hz, la frecuencia de Nyquist equivale a 60 Hz. En el caso de la señal inicial obtenida de la segunda competición de BCI [10], la frecuencia de muestreo era de 240 Hz. Por lo que la reducción en este caso es de un cuarto de la frecuencia inicial.

En segundo lugar, y con los objetivos de reducir la cantidad de fuentes de datos y de seleccionar aquellas con mayor información, se aplica PCA al conjunto de electrodos. Una vez obtenida la matriz de transformación, se seleccionan las cinco primeras dimensiones. La elección de las cinco primeras viene motivada por la suma acumulada de los cinco primeros autovalores. Como ya se introdujo en la sección 2.2.2, los autovalores permiten conocer el porcentaje de la varianza explicada en sus correspondientes dimensiones, y en este caso los cinco primeros proporcionaban suficiente información.

Una vez calculadas las nuevas dimensiones, se reestructura la señal (como se describe en la sección 2.2.1). Finalmente, y con el objetivo de evitar el implementar múltiples clasificadores, se

concatenan los periodos de señal de las cinco estructuras por tiempo. Esto es, para un mismo estímulo s , las cinco épocas registradas de longitud t en cada estructura. Como resultado se obtiene una nueva señal de longitud $5t$. En el caso anterior de la competición, el número de muestras del vector de una época, se habría visto alterado en un factor de 1,25. Esto se debe a que el vector que representa la época ha sido reducido a un cuarto de longitud (remuestreo), y luego aumentado a cinco veces esa longitud (concatenación).

Para terminar, los datos ya tratados fueron clasificados mediante una SVM. La elección de este tipo de clasificador se motiva en la sección 2.3.2, así como en las soluciones de las competiciones de BCI ya mencionadas. Conjunto con el desarrollo de la estructura para mantener la transparencia de operación de datos ya mencionada en la sección 3.1.1, se realizó una búsqueda paramétrica para determinar los valores que maximizasen los porcentajes de clasificación. Aunque tampoco se refinó mucho, debido a que este sistema se sale del objetivo principal del proyecto.

3.2. Técnicas de análisis de información

Para la extracción de las matrices de distancia o similaridad, se hizo uso de algunas de las técnicas ya introducidas en la sección 2.2.2. Más concretamente de: Correlación (máxima y estadística), SL y NCD. En esta sección se explican las peculiaridades asociadas a algunos de los métodos usados.

3.2.1. Synchronization Likelihood (SL)

Como ya se ha introducido en 2.2.3, SL es un método que permite medir la similaridad de dos series en base a sus coincidencias. Estas se definen como aquellos pares periodos con distancia menor que una cantidad ϵ de cada serie consigo misma. En este proyecto se aplica una versión modificada de este método para mejorar la afinidad con la fuente de datos.

Como ya se ha comentado en 2.2.2 y en la propia introducción a esta sección (3.1.3), las fuentes de datos a comparar resultan de la concatenación de épocas. Por tanto, en un método como SL, la iteración de la misma, puede llevar a resultados incoherentes. Por ejemplo, tomando una señal periódica cuyo ancho de ventana coincida con su periodo, este método no permite devolver resultados coherentes. Debido a que este tipo de señal siempre coincide consigo misma, siempre retornará el valor de coincidencia de la otra serie a comparar, independientemente de si realmente muestran un comportamiento similar. Por otro lado, una señal sin coincidencias consigo misma, como puede ser esta misma señal pero con tamaño de ventana mayor que la fase, devolvería como máximo su mínima coincidencia.

Para evitar esto, es posible refinar los parámetros para centrarse en un comportamiento localizado en frecuencia. Lo cual implica, la necesidad de poder localizar este comportamiento en frecuencia y que carezca de desfase de modo que siempre sea comprendido en la ventana íntegramente. En este proyecto el método aplicado elimina la iteración a lo largo de la serie, explicada en 2.2.3, para iterar en saltos de tamaño longitud de una época. De este modo, todas las mediciones de sincronización se centran en el comportamiento entre épocas.

El método fue implementado íntegramente en *Matlab*, conjunto con un método auxiliar previo que precalcula los vectores pertenecientes a cada punto de cada serie. Minimizando así, el coste total del método.

3.2.2. Distancia normalizada por compresor (NCD)

Para el cálculo de NCD, se hizo uso de la librería *Complearn*, así como el cálculo del dendrograma. Del mismo modo, se probó para los diferentes compresores incluidos dentro de la herramienta: zlib y bzlib.

Bzlib

Bzlib representa una modificación del compresor BZIP2. Este es un compresor por ordenación de bloques que hace uso de los siguientes algoritmos y métodos: Run Length Encoding (RLE), la transformada de Burrows-Wheeler (BWT), la transformada Move-To-Front (MTF) y para terminar una codificación Huffman. El proceso abreviado que sigue el algoritmo, es el siguiente:

1. **Reordenar la cadena (BWT)**: de modo que las repeticiones se maximicen manteniendo la reversibilidad. En otras palabras, transforma la información en una nueva consecución, con capacidad para volver a su forma original, que prioriza la cantidad de repeticiones de patrones de información.
2. **Transformarla en un alfabeto dinámico (MTF)**: en este caso, el objetivo es comprimir la información mediante un alfabeto. Pero con la diferencia de que este modifica en el tiempo mediante la aparición de las distintas palabras. De este modo, las repeticiones locales son fácilmente reducibles, y no se ven afectadas por su frecuencia de aparición total.
3. **Codificación en patrón-apariciones (RLE)**: de forma sencilla, transforma cada conjunto altamente repetido en una nueva expresión de *<elemento>-<número de apariciones>*.
4. **Codificación Huffman**: como reducción final de la información tratada, Bzlib usa una codificación Huffman clásica, que reinterpreta cada palabra como un conjunto de bits. Usando un criterio frecuentista, asigna representaciones más pequeñas a aquellas palabras más frecuentes (1, 01, 001, ...) y más largas a las menos frecuentes (000...01, 00...01, ...).

Zlib

La librería Zlib implementa una versión del compresor LZ77. Este hace uso de una ventana desplazada que integra un diccionario conjunto con un buffer del muestreo. De forma simple, el compresor recorre la fuente de datos intentando encontrar coincidencias entre el diccionario y 1 a n caracteres dentro del buffer. Una vez detectada la coincidencia, retorna por cada palabra encontrada una tupla. Esta contiene su posición de referencia, longitud de la palabra y carácter final de la palabra indexada. Una vez terminada esta iteración, avanza la ventana l posiciones, siendo l la distancia de la palabra, y continua hasta fin de fichero. En resumen, conforma un método de compresión adaptativa, en función de la evolución del flujo de datos de la entrada a comprimir.

En otras palabras, al igual que el diccionario dinámico de Bzlib, intenta detectar las frecuencias locales para maximizar la capacidad de compresión, con algunas diferencias. En primer lugar, al usar una ventana dinámica, no necesita reconstruir el proceso de construcción del diccionario al recuperar la cadena original. Lo cual, se compensa con tamaños de tuplas mucho mayores. Por otro lado, al hacer uso de una ventana limitada y no agrupar las apariciones de las palabras, puede perder capacidad de detección de patrones de información. Por ejemplo, cuando el tamaño de la ventana es menor que el tamaño entre aparición de patrones reconocibles.

3.3. Simulaciones

En esta sección se enumeran los sistemas desarrollados para realizar la búsqueda paramétrica sobre las variables de construcción de fichero: Número de medias y concatenaciones, y solape de construcción de series. El significado de estas en la constitución de los ficheros viene definido en la sección 3.1.3.

El sistema desarrollado se compone de: la creación de ficheros, constitución de los dendogramas, el cálculo de el coeficiente de silhouette medio y su varianza y la representación de los resultados. Todo ello se controla mediante un script en *Python* y *Bash*, para la organización de ficheros y resultados, e iteración de variables.

3.3.1. Construcción de objetos

Como se introduce en la sección 3.1.3, la construcción del formato de datos se establece en función de el promedio de ventanas, solape entre ellas, y concatenaciones de estas medias, o series. Este proceso define el formato del objeto, o fichero, a estudiar. Para poder aplicarlo sobre todo el conjunto de datos, es necesario solventar algunos problemas inherentes al conjunto de datos usado.

Al tener conocimiento de la clase asignada a cada evento, o estímulo, es posible hacer un estudio supervisado de los datos. Y aunque este enfoque es útil para la aplicación de algunas técnicas aplicadas a este problema [26], en este caso pueden llevar a conclusiones erróneas. Por ejemplo, al generar objetos de distintos estímulos pero de misma clase se producirán resultados que no representarán la realidad. Dado que los promedios tiene una componente de solape, ventana desplazada, los objetos generados siempre compartirán información, por definición, y por tanto serán más propensos a manifestar comportamientos parecidos. Por ello, se aplica un procedimiento no supervisado en la construcción de los ficheros.

Como se define en la sección 2.1.3, la profundidad del conjunto de datos puede definirse en tres niveles. En la construcción de ficheros necesaria para la simulación se establece a nivel dos, conformando subconjuntos dentro del mismo conjunto de datos. De este modo, la construcción de cada fichero solo hace uso de datos de una época de caracter, como se ha introducido previamente, perteneciente a una fila o columna (definido en profundidad en 2.1.3). En este mismo sentido, los subconjuntos definidos solo abarcan la longitud de una palabra.

Para terminar, el uso de la ventana desplazada produce errores derivados del diseño modular de construcción de datos. Al ajustarse a la cantidad de datos en cada instante, el sistema tiende a crear ficheros al final de la simulación de tamaños muy distintos al resto. Esto se debe a que en algunos casos la cantidad de datos originales, no permite producir ficheros equivalentes dada una configuración. Por lo que la generalidad adquirida en la capacidad de continuar hasta fin de datos (evitando así errores de datos no múltiplos del formato de fichero) provoca disparidad a la hora de comparar los objetos producidos. No obstante, estos últimos son eliminados de cara a los métodos de análisis de distancia, o similaridad, para evitar contaminar los resultados.

3.3.2. Cálculo de distancias y dendogramas por conjuntos

Una vez contruidos los objetos (cuya codificación se explica en la seccion 3.1.3), el cálculo de distancias se hace mediante el método de Minimun Quarter Tree Cost (MQTC) [33]. Este, conjunto con NCD se encuentra en la librería *Complearn*. MQTC, permite transformar una matriz de adyacencia pesada, en un fichero .dot. Este contiene el árbol resultante, como se explica en 2.2.3. No obstante, el uso de esta herramienta aporta un problema al desarrollo del sistema.

Para poder obtener un árbol con mayor calidad, la herramienta intenta minimizar una unidad de error. A mayor cantidad de nodos, más difícil resulta minimizar el error, debido al alto coste computacional [33]. En el caso de una simulación es necesario un tiempo razonable por operación básica, por lo que es necesario obligar al sistema a escoger un grafo de menor calidad. De lo contrario una operación básica podría tener una duración de horas, haciendo la simulación inviable. Para reducir la calidad del grafo, es necesario minimizar el número de nodos usados. Esto, en algunos casos, implica la construcción de un grafo no representativo. De este modo se presenta el problema de elegir ente un grafo representativo o de calidad.

La solución propuesta en este proyecto escoge un subconjunto de nodos (objetos u archivos) balanceada entre las clases "sí" y "no" (supervisado, con P300 y sin P300) de forma aleatoria. Este proceso se repite varias veces por iteración para, en promedio, obtener un término intermedio entre calidad y representatividad del grafo con respecto a la muestra original.

3.3.3. Búsqueda paramétrica: Coeficiente de Silhouette

Aunque la representación en forma de grafo es útil para detectar a simple vista la capacidad de clusterización de los nodos, es necesario establecer un criterio automático de cara a la simulación. Para ello se hizo uso del Coeficiente de Silhouette, descrito en la sección 3.1.3. Este coeficiente permite medir la calidad de los clusters incluidos en un grafo. Y en última instancia la calidad de los clusters de cada una de las clases.

En el desarrollo de la simulación, se aplica esta métrica sobre cada conjunto de grafos generados para cada iteración. Registrando un coeficiente para cada clase y otro para el total, promediados en tres matrices respectivamente. Del mismo modo, se construye otras tres matrices con las varianzas de cada uno, con respecto al conjunto correspondiente. La razón de incluir la varianza en la representación final, es la de comprobar que los resultados obtenidos no son fruto de la aleatoriedad establecida.

Para medir la distancia en tiempo en la que se mantiene la clusterización, se aplican sobre diferentes subconjuntos de datos. De modo que cada uno de los conjuntos abarca una longitud de tiempo diferente y una estacionaridad distinta.

3.3.4. Representación de resultados

Para mostrar los resultados, se establece un sistema de representación mediante mapas de color, para estudiar la tendencia de cada simulación. En cada simulación se produce un fichero tipo csv, con los resultados de la varianza y medias de los coeficientes de Silhouette [39] de cada iteración. Estos incluyen los coeficientes medios para cada clase y para el conjunto total, sobre cada conjunto (palabra), y sobre un conjunto y parte del siguiente. Esto se explica más adelante como toma de contacto de sistema de clasificación, y se basa en "invadir" el grafo de un conjunto con nodos de la clase del siguiente. La hipótesis para la realización de esta prueba es la de intentar aprovechar la continuidad de la estacionaridad (o parte) local para clasificar de forma adaptativa.

3.4. Conjuntos de Datos

La fuente de datos utilizada, pertenece a la segunda competición de BCI. La razón de seleccionar esta fuente de datos, fue para poder hacer uso de un conjunto controlado y resuelto. Debido a la gran cantidad de dificultades derivadas de la naturaleza del problema, resulta conveniente minimizar aquellas provenientes del conjunto de datos.

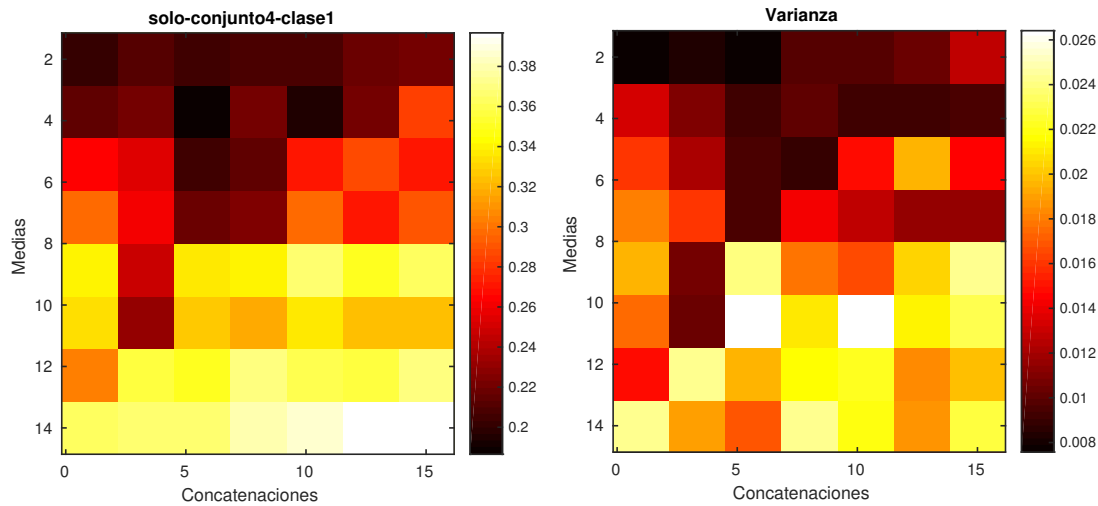


Figura 3.3: Sistema de representación de resultados, mediante mapa de color. Los ejes de las matrices representan los valores de configuración como se define en 3.3.1. a) La primera matriz representa el coeficiente de Silhouette obtenido. b) La segunda representa la varianza asociada a cada valor

3.4.1. Conjunto de Datos 1: BCI Competition II

Publicada en Noviembre el 2002, el objetivo de esta competición es el de detectar los periodos de P300 de un conjunto de prueba, dado uno de entrenamiento. La técnica de adquisición fue EEG con una frecuencia de muestreo de 240 Hz y 64 canales. En el conjunto de entrenamiento se definen diferentes subconjuntos, que en este proyecto se han diferenciado en "niveles", ya introducidos en la sección 2.1.3. Este conjunto se divide de la siguiente forma:

Sesion 10

- "CAT"
- "DOG"
- "FISH"
- "WATER"
- "BOWL"

Sesión 11

- "HAT"
- "HAT"
- "GLOVE"
- "SHOES"
- "FISH"
- "RAT"

En la simulación se hizo uso solo de un canal con resultados conocidos (El canal 11 correspondiente con el Cz). Además, se enumeraron los conjuntos de la sesión 11 del 1 al 6 para su uso automático como se ha indicado en la sección 3.3.4.

Soluciones En las soluciones enviadas a la competición se alcanza el 100 % de clasificación en varias de ellas. Por lo tanto representa un conjunto de datos sencillo. Los resultados se pueden apreciar en la tabla 3.1

#.	Contribuidor	Error	Repeticiones
1.	Matthias Kaper	0.0 %	5
1.	Xiaorong Gao	0.0 %	5-8
1.	Vladimir Bostanov	0.0 %	6
1.	Benjamin Blankertz	0.0 %	6-11
1.	David Tax	0.0 %	n/a
6.	Justin Werfel	54.8 %	n/a
7.	Elena Glassman	64.5 %	n/a

Tabla 3.1: Tabla de resultados de la segunda competición de BCI

4

Experimentos Realizados y Resultados

En las secciones anteriores, se ha presentado algunos de los sistemas implementados en el desarrollo de este proyecto de fin de Máster. En primer lugar, se presentó la estructura de datos genérica, usada de forma común a lo largo del proyecto, aportando transparencia al manejo de los datos. La razón principal era que el intervalo tras estímulo, tenía una duración mayor al periodo entre estímulos, generando un solape. También, la necesidad de acceso eficiente a los datos, fue otra razón para el diseño de esta estructura. A parte, por supuesto, de la facilidad de uso de cara al usuario una vez estructurado. La estructura comprendía las diferentes ventanas de señal asociadas a un evento, y su aparición en el tiempo relativa a su estímulo. Del mismo modo, esta estructura permanecía dependiente del electrodo escogido como fuente de datos.

En segundo lugar, se revisaron y usaron diferentes técnicas, con el objetivo de medir la similitud existente entre un determinado numero de fuentes. En este caso, y haciendo uso de la estructura ya mencionada, se definió un formato de datos, así como una estructura de objeto o fichero. El objetivo de esta comparación, es encontrar parecidos entre aquellas técnicas asociadas a la clase con el evento P300 asociado. Mientras que, los objetos restantes no deberían de mostrar mayor parecido que el propio ruido de la señal. El resultado de estas pruebas devuelve una matriz de distancias, o similitud, del parecido de cada objeto con todos los demás.

Para poder estudiar estos resultados, se escogió una representación en forma de árbol, haciendo uso de un algoritmo de construcción que permitiese discriminar entre rangos de valores muy ajustados de manera óptima. De este modo, la representación en forma de grafo aportaba la capacidad de evaluar visualmente la separación clusterización de las distintas técnicas de similitud. No obstante, para no depender de valoraciones relativas a la esta evaluación del grafo, se estableció una métrica formal de calidad de cluster. Entendiendo estos como el par de subconjuntos de nodos hojas pertenecientes a cada una de las dos clases.

Gracias a las automatizaciones realizadas en la construcción de los objetos, o ficheros, así como la métrica de evaluación de cluster, permitieron diversas simulaciones. Por ejemplo, la búsqueda paramétrica de las variables asociadas a la construcción del fichero. Adicionalmente, se desarrolló un clasificador simple en función de los datos obtenidos de la segunda competición de BCI, con el objetivo de usarlo más tarde como base.

En esta sección se desarrollaron pruebas con el objetivo de probar la funcionalidad básica de los sistemas implementados y experimentos para estudiar su comportamiento sobre fuentes de datos reales.

4.1. Pruebas de validación: Ganando intuición

El objetivo de esta sección es el de establecer una serie de pruebas, con conocimiento previo del resultado esperado, para comprobar la eficacia de los distintos métodos. Y adicionalmente, ganar una intuición sobre la aplicación de métodos estudiando su rango de resultados conforme a las distintas pruebas. Para ello se hizo uso de datos sintéticos comunes para todos los métodos usados en los experimentos finales. Usando como funciones principales senos y ruido gaussiano blanco(wgn). A continuación, se incluye la descripción de cada una de las pruebas:

1. $\sin(x) - \sin(x)$: La identidad, que debe de corresponder al valor de similaridad máxima.
2. $\sin(x) - \sin(x + \pi)$: Desfase, que altera la señal, pero manteniendo una relación de comportamiento.
3. $\sin(x) - k \sin(x)$: Modificando la amplitud, pero con misma fase y comportamiento.
4. $\sin(x) - wgn() + \sin(x)$: Agregando una función de ruido gaussiano con media 0 y varianza 1.
5. $wgn() - \sin(x)$: Eliminando cualquier patrón determinado para minimizar el resultado de similaridad, más allá del aleatorio.
6. $wgn() - wgn()$: Igual que la anterior, pero sin ningún comportamiento determinado.

Los resultados de las pruebas realizadas se muestran en la siguiente tabla 4.1, donde se puede apreciar las salidas producidas por los distintos métodos. Entre los resultados obtenidos, es posible apreciar que en todas las técnicas, la identidad se reconoce correctamente. En el caso de correlación máxima (MaxCorr), no se retorna un valor entre 0 y 1, debido a que el valor no está normalizado. No obstante, al compararlo de forma relativa entre las pruebas realizadas sobre este método, este resulta suficiente.

Analizando los resultados de las pruebas 2 y 3, se aprecia que mientras que correlación máxima y SL ignoran las separaciones en fase de las señales, penaliza en gran medida los cambios en la amplitud de esta. Justo al contrario que la correlación estadística (StdCorr). NCD por otro lado, puntua por igual, negativamente, a todas las pruebas desde la 2 a la 5, identificando sin embargo, pequeños parecidos en los conjuntos aleatorios de ruido.

Respecto a la prueba 4, todas reducen el valor de relación a la mitad, aproximadamente, mientras que NCD decae drásticamente entorno a 10^{-3} . Por lo que cabría esperar, que NCD se presente más estricto respecto al ruido agregado a la señal.

En ultimo lugar, los resultados de la prueba 5, en los que el valor debería de ser mínimo, debido a que se trata de series sin atributos relacionados, muestran que SL detecta parecidos entre dos fuentes distintas. Esto se debe, posiblemente, a que SL pondera los pares de coincidencias entre una misma señal y la otra.

4.2. Simulaciones sobre parámetros

En esta sección se muestran los resultados obtenidos de algunas de las distintas pruebas realizadas durante el desarrollo de este proyecto, más concretamente sobre las búsquedas paramétricas o simulaciones. Cada prueba es definida como un conjunto de método aplicado y un conjunto de datos, que se especifican en el título del apartado de cada prueba.

Método	Prueba 1	Prueba 2	Prueba 3	Prueba 4	Prueba 5	Prueba 6
MaxCorr	9.999e+3	1.111e+2	9.999+3	5.6744e+3	1.33e+2	3.082e+2
StdCorr	1.	0.0036	1	0.5675	-0.0024	0.0111
SL	0.9974	0.9436	0.3325	0.6102	0.7752	0.6076
NCD (bzlib)	1.	0.	0.004781	0.00694	0.007637	0.010472
NCD (zlib)	1.	-0.000856	0.004782	0.00694	0.007637	0.010472

Tabla 4.1: Resultados obtenidos por los distintos métodos para cada una de las pruebas de validación realizadas. MaxCorr: máxima correlación, StdCorr: Correlación estadística, SL: Synchronization Likelihood, NCD: Distancia normalizada por compresor.

4.2.1. Estructura de las simulaciones

Las pruebas realizadas para los métodos introducidos sobre los distintos conjuntos de datos, siguen un formato común. Este es introducido brevemente en la sección 3.3.4. El proceso interno del transcurso de cada prueba se divide en:

- **Busqueda paramétrica**, que alterna los valores de configuración del sistema.
- **Iteración sobre una configuración**, evitando así los problemas relacionados con grandes conjuntos de datos
- **calculo de distancias**, que constituye el proceso atómico de generación y cálculo de calidad de cluster sobre grafos

Búsqueda paramétrica

La búsqueda paramétrica, se desarrolla de forma regular sobre el espectro de valores especificado (sin profundizar bajo ningún criterio). Esto es interesante para probar la eficacia de la composición del objeto (estructura o fichero de datos) en la clusterización del sistema. Por ello, se representa como un mapa de color de los promedios de los coeficientes de Silhouette obtenidos en el paso inferior. Del mismo modo, se representa su varianza para analizar más rigurosamente la información.

Iteración sobre una configuración

El siguiente nivel de la simulación, conforma el cálculo de los promedios y varianzas de los coeficientes de Silhouette. Este sistema, constituye una solución al problema introducido en la sección 3.3.2. Este se fundamentaba en la incapacidad del algoritmo de generación de grafos usado (*Maketree*) para manejar grandes cantidades de nodos, debido a su naturaliza de problema NP. Para poder obtener una métrica fiable y representativa, sin perder calidad en la representación del grafo, se opta por el cálculo sobre subconjuntos de nodos (aleatorios) de forma iterada. De este modo, al obtener grafos de mayor calidad, el coste computacional se reduce en gran medida y al usar diferentes iteraciones para el cálculo de los grafos, el coeficiente obtenido es más regular.

Cálculo de distancias

Para terminar, el proceso de cálculo de grafos y calidad de clusters, posee gran complejidad. En primer lugar, el cálculo de los grafos se divide en dos subsistemas: compresores y resto de métodos. El primero, genera todos los ficheros necesarios al principio de su ejecución, y tras organizarlos, recorre todos los conjuntos ejecutando el mismo cálculo sobre todos los subconjuntos. Este método presenta dos puntos críticos en la ejecución del mismo: la generación de ficheros inicial, que conlleva un coste muy alto antes de iniciar la simulación (dependiente de la velocidad de disco); y la convergencia del grafo, como ya se ha comentado, debido a que el algoritmo necesario (explicado en la sección 2.2.3) necesita iterar configuraciones para reducir el error del grafo producido. Este último problema es común para ambos subsistemas. Se soluciona agregando una espera, que de ser superada, produce la cancelación de la búsqueda, dejando el último grafo probado, y anotando un "fallo", como información útil en el estudio. Este subsistema, es usado por los métodos con base en compresión.

El segundo subsistema aplicado, pertenece al resto de métodos aplicados, y sustituye la generación de ficheros, por una estructura en memoria. Esto mejora la velocidad, pero reduce la operabilidad para casos futuros dado que los datos no se guardan para cada iteración. Del mismo modo, la generación de matrices de distancia es total, y no parcial como en el anterior caso. Con lo que el sistema, necesita de un muestreo añadido de estas matrices para producir los subconjuntos de grafos necesarios para el cálculo de promedios. Este coste agregado, sumado a la ineficiencia de los métodos usados, incrementa el coste sustancialmente con respecto al caso anterior. Ambos métodos guardan todos los grafos generados, así como las matrices de distancia y los mapas de color (en forma de ficheros .csv), por lo que pueden ser consultados y revisados tras la simulación.

4.2.2. Escenarios de pruebas

Dentro de las simulaciones realizadas es posible distinguir conjuntos de pruebas en función de los parámetros a iterar. De este modo, se definen los siguientes grupos: Pruebas basadas en la estructura de los datos y de selección de electrodos. En las pruebas sobre la estructura datos, se obtienen resultados sobre todas las técnicas implementadas, para observar su comportamiento ante la evolución de la esta. De este modo, es posible ganar intuición ante el ajuste del método ante la estructura de la información, y ver como evoluciona en conjunto con esta. Todas las pruebas iteradas sobre la estructura, se realizaron sobre el conjunto de datos.

Por otro lado, las pruebas de valoración de electrodos resultan una toma de contacto hacia el uso de los sistemas desarrollados en este proyecto, en la implementación de una primera aproximación hacia la adaptabilidad. En este caso, los datos se analizan en los subconjuntos o palabras definidos en 3.4.1.

4.2.3. Simulaciones sobre parámetros de estructura del evento

En esta sección se muestran los resultados obtenidos de la ejecución del proceso explicado en la sección 4.2 con los distintos métodos implementados. Los conjuntos de datos, a su vez, se presentan divididos en subconjuntos, o palabras, enumeradas de acuerdo a lo descrito en la sección 3.4.1. Cada simulación comprende la ejecución de uno de estos subconjuntos, mediante la técnica seleccionada. No obstante, y como aproximación hacia un sistema adaptativo, se incluye una variante en los resultados de las primeras pruebas. La modificación añadida, se ha desarrollado mediante la inclusión de datos de la clase asociada al evento P300 del subconjunto siguiente al seleccionado. La idea detrás de esta prueba, es la de comprobar si la calidad del cluster se mantiene, o mejora, entre palabras distintas (subconjuntos), de modo que pueda usarse en el

futuro como método propio de clasificación. A esta aproximación se la ha denominado "invasión" de grafo, debido a la base de hacer uso del grafo previo como base para ganar información. No obstante, al conllevar una dificultad agregada al progreso de la simulación, esta prueba adicional solo se ha realizado para algunas pruebas.

Las pruebas realizadas, recogen los datos de calidad de cluster y varianza del mismo, para cada palabra, así como para el método de clasificación propuesto en la sección 3.1.3. El objetivo es el de perder incertidumbre sobre la viabilidad de esta propuesta, de cara a una posible investigación futura. Para reducir la extensión de cada apartado, y con el fin de hacer más cómoda la revisión de los resultados, estos se resumen en unas breves conclusiones, junto con un ejemplo de la simulación (el mismo caso para todas las pruebas) y dos tablas resumen sobre los datos obtenidos. A modo de resumen cada prueba se compone de:

1. Matriz de resultados de la simulación sobre el subconjunto 4.
2. Matriz de varianza asociada a la simulación sobre el subconjunto 4.
3. Tabla resumen de los valores máximos, medianas y mínimos totales, sobre todos los subconjuntos.
4. Tabla resumen de los valores máximos, medianas y mínimos totales, de clase aislada los subconjuntos.
5. Ejemplo de dendrograma de la simulación.

Adicionalmente en algunas pruebas se incluye también las pruebas de invasión de grafo, previamente comentadas, y figuras adicionales para explicar sucesos particulares de la prueba.

I) Bzlib (NCD) : Conjunto A

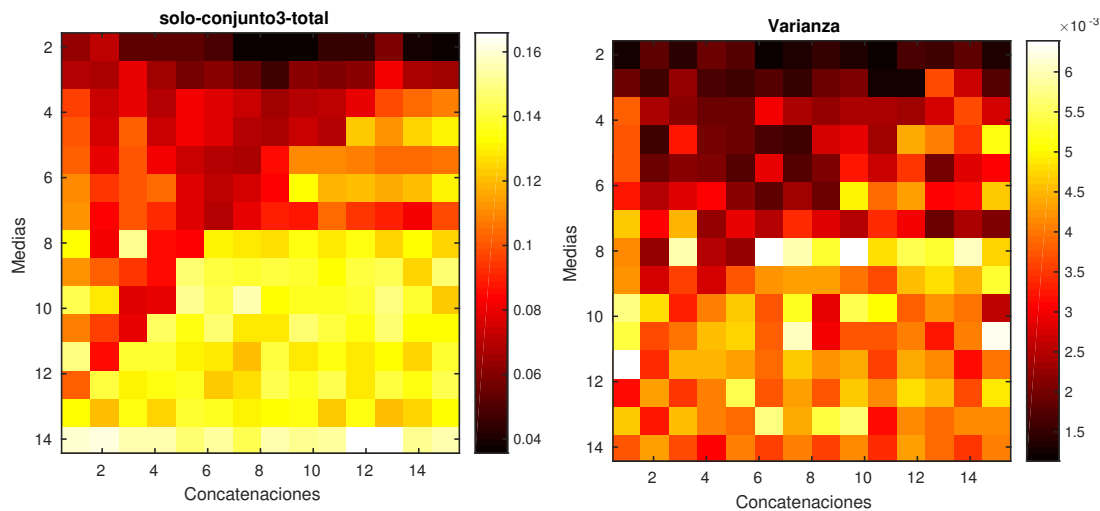


Figura 4.1: Mapas de color resultado de la prueba I (bzlib), sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)

En los resultados obtenidos sobre el primer conjunto de datos (3.4.1), haciendo uso de NCD con el compresor Bzlib, se puede observar un incremento en la clusterización del grafo en gran medida según avanza la simulación. La mayoría de los ejemplos muestran esa tendencia en

ambas variables como se muestra en el ejemplo de la figura 4.1. Aunque este es solo un caso de los numerosos subconjuntos recorridos, es posible observar la tendencia de mejora según se incrementan ambas variables.

Dentro del peso otorgado por cada variable, el número de medias usadas tiende a tener más importancia en la iteración de la simulación hasta el punto de hacer casi inocuo el cambio de la otra variable. Esta tendencia muestra la importancia a la hora de construir la estructura de datos, con respecto al uso de Bzlib. Por otro lado, la convergencia observada al final de la simulación, se debe a que, dado un gran número de medias, no queda información suficiente como para poder concatenar las diferentes series. Este hecho se introducía en la sección 3.1.3.

Set	Máximo	Mínimo	Mediana
1	0.172256	0.045364	0.091244
2	0.161057	0.032669	0.102609
3	0.171699	0.031367	0.110561
4	0.166846	0.035638	0.114347
5	0.150856	0.037497	0.106679
1&2	0.161159	0.029060	0.095786
2&3	0.168415	0.026116	0.101742
3&4	0.170566	0.031300	0.116480
4&5	0.150802	0.053751	0.095568
5&6	0.147956	0.051189	0.092285

Tabla 4.2: Resultados máximo, mínimo y medio obtenidos para cada simulación sobre el primer conjunto de datos con Bzlib para cada subconjunto

Como se adelantaba en la introducción de esta sección, también se incluyen los resultados referentes a la propuesta de clasificación mediante grafo. En estas pruebas, se hace uso del subconjunto de datos completo de la palabra previa, y se introducen los datos de la clase del evento a detectar, del siguiente. Teniendo como objetivo mantener la calidad de cluster. En los resultados obtenidos. Mostrados en la segunda parte de la tabla 4.2, es posible observar que la variación de esta no es significativa. Lo que apoya la hipótesis para el desarrollo de un método de clasificación valorando métricas referentes al cluster, como cercanía o incluso el incremento o decremento de la calidad del mismo.

Para terminar, y como ya se adelantaba en 2.2.2, la naturaleza de este problema hace difícil identificar dos clases distintas mediante medidas de similaridad debido a que solo una de ellas puede compartir información. La otra, es identificada por descarte respecto a la primera, y las propiedades que esta representa no son interesantes. Esto es así, debido a que el objetivo final es el de, dentro de una misma época, poder discriminar un subconjunto de datos pertenecientes a la primera clase frente a los de la segunda. En otras palabras, una clasificación por votación. Por tanto, la idea de medir la calidad "parcial" de los clusters toma fuerza. Esta se muestra en la tabla 4.3, con la misma estructura que la tabla 4.2 solo que separando entre el cluster de cada clase.

A modo de ejemplo, uno de los grafos asociados a la figura 4.1 puede se muestra en la figura 4.2. No obstante, en la sección A se muestran más figuras obtenidas del desarrollo de las diversas simulaciones.

II) Zlib (NCD) : Conjunto A

En primer lugar, la resolución de tanto de esta como del resto de pruebas, se redujo durante la realización del proyecto, debido al alto coste computacional total de la simulación. Y dado

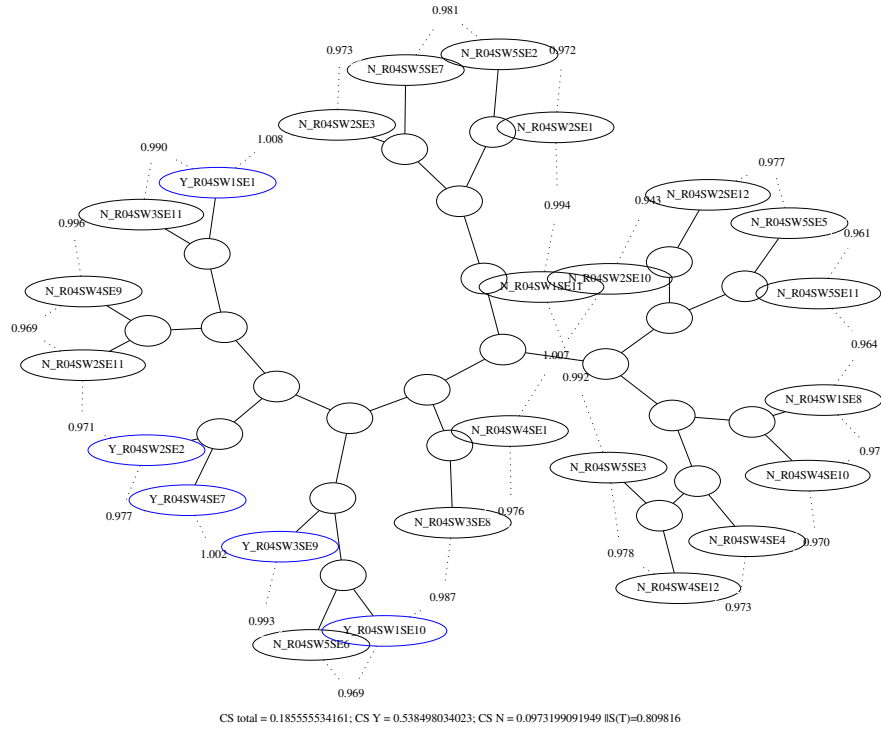


Figura 4.2: Ejemplo de dendrograma obtenido en la simulación sobre el primer conjunto de datos con el método bzlib.

Clase A				Clase B		
Set	Máximo	Mínimo	Mediana	Máximo	Mínimo	Mediana
1	0.403312	0.122108	0.239574	0.124153	0.010173	0.056563
2	0.406395	0.169071	0.251505	0.112605	-0.007656	0.060295
3	0.419219	0.177863	0.279348	0.113467	-0.016275	0.066978
4	0.396725	0.186530	0.292511	0.112331	-0.010824	0.067718
5	0.374970	0.185461	0.278285	0.097871	-0.008312	0.065213
1&2	0.387618	0.183456	0.245241	0.113496	-0.017878	0.057655
2&3	0.418533	0.178586	0.266811	0.109762	-0.028102	0.063060
3&4	0.403610	0.182873	0.288011	0.114072	-0.018719	0.066726
4&5	0.383126	0.186204	0.288471	0.103003	-0.011227	0.066031
5&6	0.370494	0.191490	0.259439	0.095203	0.015207	0.055665

Tabla 4.3: Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con Bzlib. A diferencia de la tabla 4.2, en esta tabla se muestran los coeficientes para cada clase por separado.

que la motivación de estas, es la de observar la tendencia y valores máximos de la misma, no es necesario el aplicar un grado de resolución mayor.

Respecto a los resultados obtenidos, el compresor Zlib mantiene la tendencia y capacidad de mejora observada en la primera prueba. Aunque es cierto, que los valores máximos no se ven afectados en gran medida, el promedio de la simulación si que resulta ser menor que el anterior. No obstante, esto puede deberse a la reducción de resolución de la simulación ya comentada. Otro factor a observar, es la diferencia de calidad de clusters entre clases. En la primera prueba la clase B es mejor detectada que la A, con respecto a esta. De este modo, la capacidad de ajustar mejor la clase objetivo de Zlib, se compensa con una detección general mejor del sistema

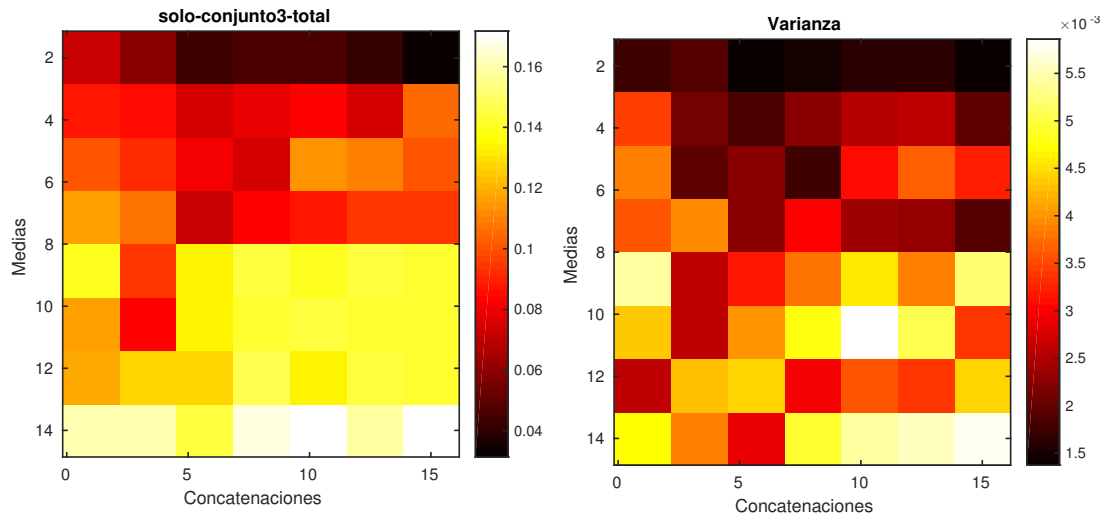


Figura 4.3: Mapas de color resultado de la prueba II, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)

de Bzlib. Esto puede ser interesante de cara a la aplicación final, como puede ser la de centrarse solo en la calidad de la clase de los eventos relevantes.

Set	Máximo	Mínimo	Mediana
1	0.172302	0.051139	0.083807
2	0.164995	0.054016	0.093432
3	0.161025	0.052610	0.097884
4	0.158719	0.056001	0.098568
5	0.146312	0.054557	0.092359
1&2	0.166512	0.048811	0.090379
2&3	0.160507	0.049864	0.094760
3&4	0.164250	0.052740	0.098804
4&5	0.150802	0.053751	0.095568
5&6	0.147956	0.051189	0.092285

Tabla 4.4: Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con Zlib para cada conjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior

III) Correlación cruzada (máximo) : Conjunto A

Al contrario que el apartado anterior, en esta prueba el método usado no hace uso del formato de datos descrito 3.3.1, necesario para las técnicas basadas en compresores. En este caso, la fuente de datos solo se construye en base a las concatenaciones y medias, pero en su representación numérica. Esto se explica con mayor detalle en la sección 3.1.3.

Los resultados obtenidos en las sucesivas simulaciones, muestran una drástica mejora en la calidad de clusterización del sistema, para cierta relación entre las concatenaciones y las medias. Al igual que en las dos pruebas anteriores aparece una frontera en la matriz. Esta resulta un umbral o mínimo para generar grafos con clusters adecuados. No obstante, en esta prueba los valores obtenidos presentan unos cambios más bruscos y su capacidad de incremento parece limitada.

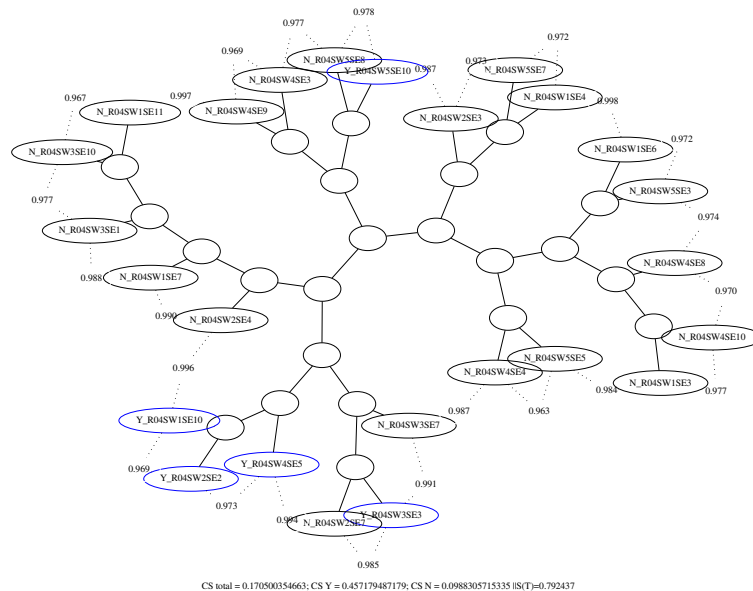


Figura 4.4: Ejemplo de dendrograma obtenido en la simulación sobre el primer conjunto de datos con el método zlib.

Clase A				Clase B		
Set	Máximo	Mínimo	Mediana	Máximo	Mínimo	Mediana
1	0.444473	0.136151	0.257041	0.106763	0.012122	0.045231
2	0.400993	0.168553	0.249453	0.107465	0.019361	0.051780
3	0.391614	0.180910	0.264370	0.103590	0.017625	0.058565
4	0.377604	0.190495	0.265098	0.108716	0.019861	0.058662
5	0.361845	0.189871	0.258435	0.098318	0.017856	0.056463
1&2	0.401642	0.181098	0.247883	0.113231	0.013711	0.050410
2&3	0.389241	0.180312	0.253638	0.105081	0.012508	0.052398
3&4	0.384060	0.189105	0.264207	0.109297	0.017922	0.058383
4&5	0.366625	0.193764	0.262152	0.102805	0.017652	0.059114
5&6	0.370494	0.191490	0.259439	0.095203	0.015207	0.055665

Tabla 4.5: Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con Zlib. A diferencia de la tabla 4.4, en esta tabla se muestran los coeficientes para cada clase por separado.

Por otro lado, los máximos obtenidos son mayores, aunque con una media sensiblemente más baja. Cabe destacar también, que solo aquellas configuraciones con resultados buenos, retornan una varianza significativa con respecto al resto de puntos. En la la tabla 4.7, continua esta tendencia, con máximos superiores y medias inferiores.

IV) Correlación estadística : Conjunto A

Los resultados obtenidos para la aplicación de correlación estadística, resultan ser muy similares a los mostrados en la prueba de correlación máxima. Además, aunque inicialmente se aprecian cambios en puntos aislados de la simulación, la tendencia general es idéntica a la mostrada en el apartado anterior. No obstante, los máximos obtenidos en este caso, son menores que los mostrados en el caso menor 4.8. Y aunque la diferencia no es muy grande, si que se mantiene

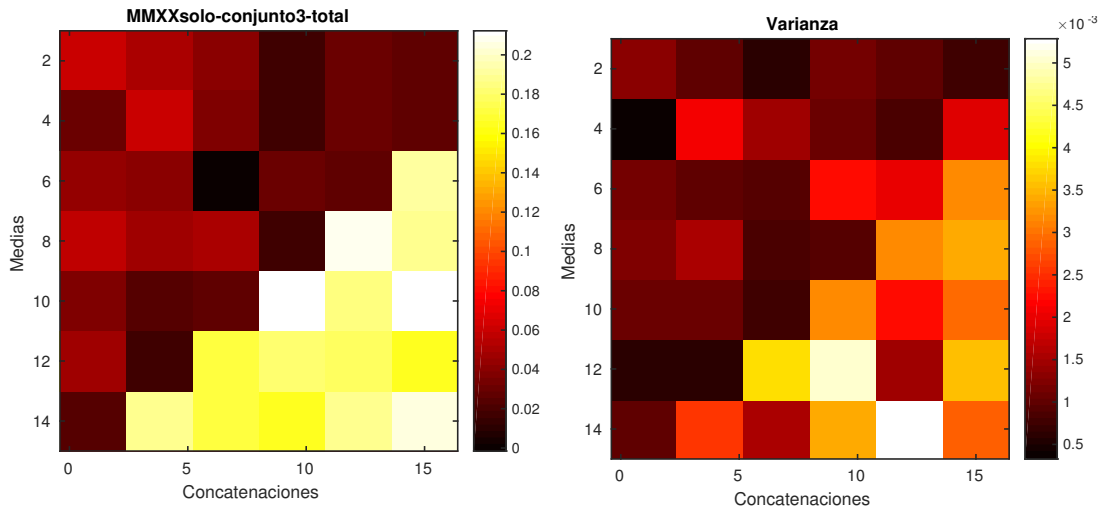


Figura 4.5: Mapas de color resultado de la prueba III, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)

Set	Máximo	Mínimo	Mediana
1	0.223043	0.002995	0.062284
2	0.226444	-0.001857	0.056112
3	0.211655	0.001822	0.062526
4	0.221697	0.006912	0.062470
5	0.227648	0.004069	0.060693

Tabla 4.6: Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con correlación máxima para cada subconjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior

Clase A				Clase B		
Set	Máximo	Mínimo	Mediana	Máximo	Mínimo	Mediana
1	0.563828	0.143267	0.259521	0.177832	-0.075126	0.026069
2	0.539692	0.144621	0.260636	0.175511	-0.082711	0.014880
3	0.487341	0.153173	0.252832	0.158572	-0.071824	0.025750
4	0.488657	0.146684	0.259386	0.173577	-0.06627	0.024425
5	0.485524	0.144727	0.251606	0.178443	-0.066142	0.023315

Tabla 4.7: Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con máxima correlación. A diferencia de la tabla 4.6, en esta tabla se muestran los coeficientes para cada clase por separado.

en todos los casos.

Por contrapartida, la varianza media de las simulaciones de esta prueba, es levemente superior a la de la técnica previa. Lo cual, aun siendo la diferencia leve, si que permite decantarse por el uso de correlación máxima sobre correlación estadística para este tipo de problema.

V) Synchronization Likelihood : Conjunto A

Para terminar dentro de las técnicas de similitud, se estudió el conjunto de datos mediante SL. Aunque los resultados mostrados en la tabla 4.10 muestran unos máximos en la clasificación

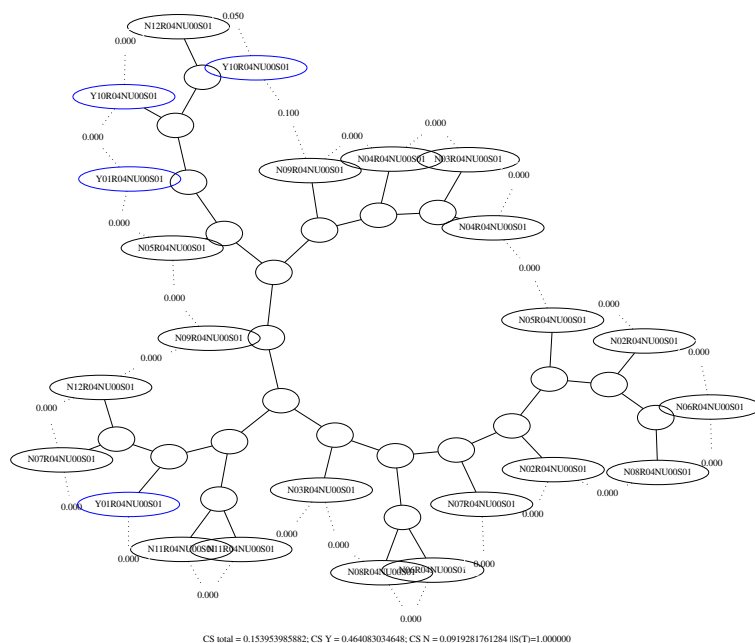


Figura 4.6: Ejemplo de dendrograma obtenido en la simulación sobre el primer conjunto de datos con el método de máxima correlación.

Set	Máximo	Mínimo	Mediana
1	0.209432	-0.005293	0.062356
2	0.223002	-0.001917	0.060941
3	0.231939	0.002410	0.066487
4	0.219745	0.003699	0.066547
5	0.219156	-0.005768	0.036760

Tabla 4.8: Resultados máximo, mínimo y medio de clusterización total por cada simulación del primer conjunto de datos con máxima correlación para cada subconjunto. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior

Clase A				Clase B		
Set	Máximo	Mínimo	Mediana	Máximo	Mínimo	Mediana
1	0.535302	0.143812	0.274007	0.163139	-0.089034	0.022572
2	0.513294	0.167372	0.281140	0.173957	-0.081516	0.022867
3	0.519487	0.173810	0.247910	0.199377	-0.075114	0.028187
4	0.502720	0.163206	0.258872	0.173691	-0.067978	0.027901
5	0.533569	0.185977	0.237320	0.156273	-0.091045	-0.017533

Tabla 4.9: Resultados parciales obtenidos durante las simulaciones del primer conjunto de datos con correlación estadística. A diferencia de la tabla 4.8, en esta tabla se muestran los coeficientes para cada clase por separado.

muy parecidos a los ya observados en las pruebas anteriores, la tendencia de la simulación destaca el poco efecto de esta el formato de datos sobre el problema con este método. La ausencia de progreso alguno, o respuesta ante diferencias de información, son significativos para concluir que este tipo de técnicas no permiten capturar la información del mismo modo que los demás del problema.

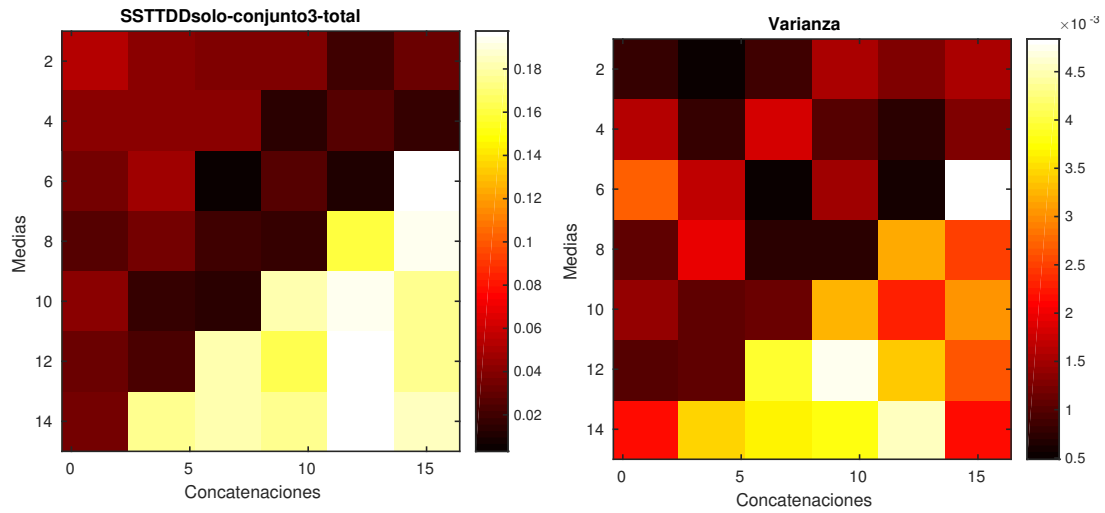


Figura 4.7: Mapas de color resultado de la prueba IV, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)

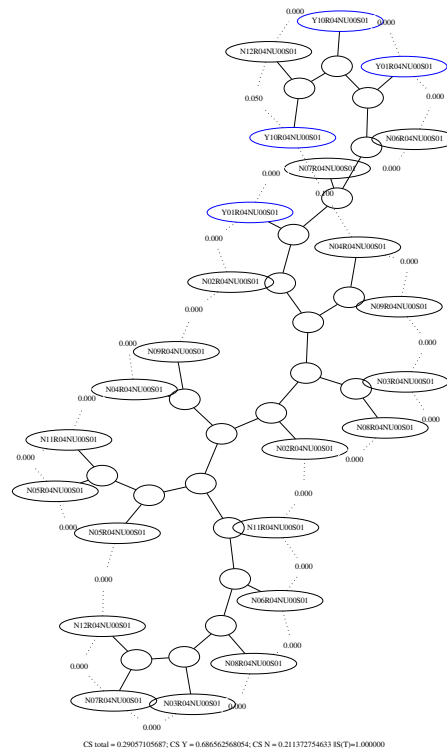


Figura 4.8: Ejemplo de dendrograma obtenido en la simulación sobre el primer conjunto de datos con el método de correlación estadística.

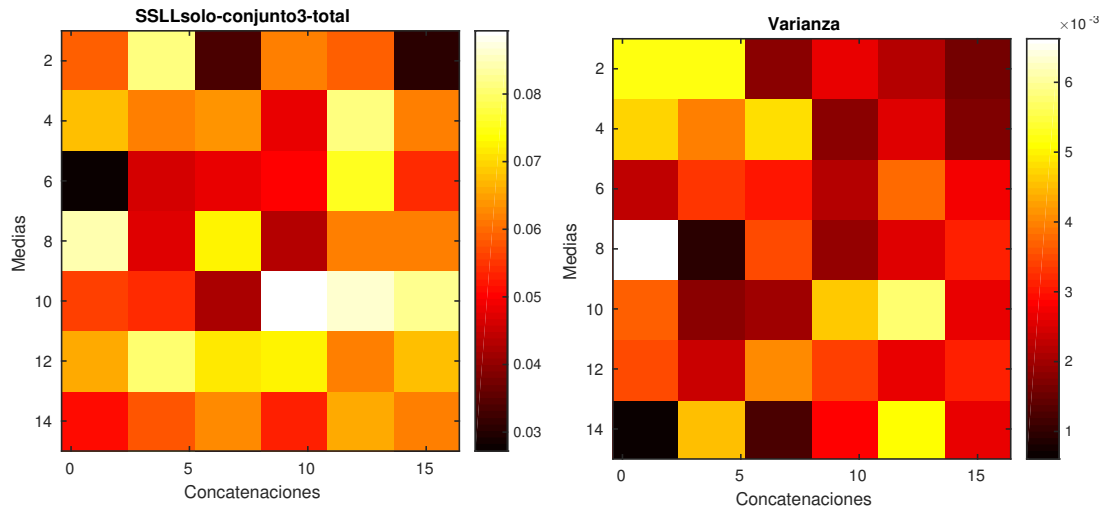


Figura 4.9: Mapas de color resultado de la prueba V, sobre el subconjunto 4. a) La primera matriz muestra el coeficiente de Silhouette obtenido en cada iteración. b) Muestra la varianza correspondiente a cada punto de la figura a)

Set	Máximo	Mínimo	Mediana
1	0.242030	0.005409	0.064650
2	0.225049	0.004122	0.058664
3	0.233325	0.001717	0.061964
4	0.233424	0.014794	0.063954
5	0.248456	0.001937	0.067705

Tabla 4.10: Resultados máximo, mínimo y medio de clusterización total por cada simulación de cada subconjunto del primer conjunto de datos con SL. Se incluye los resultados obtenidos en la invasión de grafos en la tabla inferior

Así mismo, al consultar los árboles obtenidos en el proceso, se puede observar la anomalía que permite obtener unos valores tan altos en el cómputo de calidad del cluster de la clase, en la figura 4.11. Debido a los coeficientes carentes de estructura aportados por SL, el árbol generado tiende a mostrar una topología en forma de línea. Lo cual beneficia en gran medida al cálculo del coeficiente de Silhouette.

Clase A				Clase B		
Set	Máximo	Mínimo	Mediana	Máximo	Mínimo	Mediana
1	0.526499	0.141562	0.307943	0.199065	-0.077091	0.036280
2	0.569991	0.173663	0.272056	0.181363	-0.074328	0.019609
3	0.488944	0.134707	0.255013	0.187694	-0.068396	0.026015
4	0.526667	0.166350	0.289568	0.186473	-0.054135	0.02411
5	0.511717	0.164415	0.275650	0.195804	-0.072598	0.028642

Tabla 4.11: Resultados parciales obtenidos durante la simulaciones del primer conjunto de datos con SL. A diferencia de la tabla 4.10, en esta tabla se muestran los coeficientes para cada clase por separado.

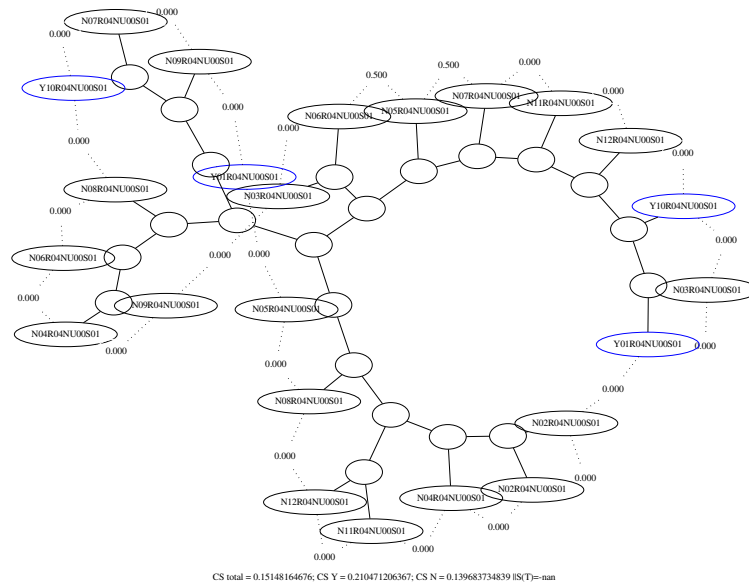


Figura 4.10: Ejemplo de dendrograma obtenido en la simulación sobre el primer conjunto de datos con el método SL.

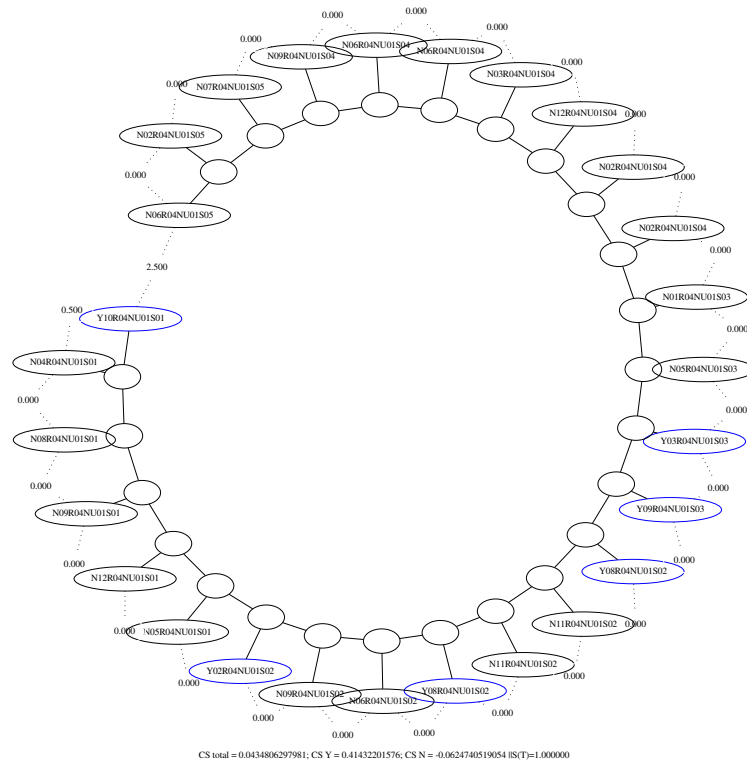


Figura 4.11: Otro ejemplo de dendrograma obtenido con SL. En este caso se observa la anomalía producida por la baja adaptación del método a la estructura del evento

4.2.4. Puntuación de electrodos

En esta sección se introduce la idea del desarrollo de otro método posible de adaptación haciendo uso de los diferentes métodos usados a lo largo de este proyecto. El objetivo es aprovechar el coeficiente de calidad de cluster proporcionado por el sistema para una configuración, y usarlo para establecer una puntuación para cada electrodo. De este modo, podría desarrollarse un algoritmo de selección que confrontase las distintas puntuaciones de cada electrodo en el tiempo, o como el subconjunto común de diferentes métricas para obtener aquellos mejor valorados.

Para ganar intuición con respecto a estas propuestas de desarrollo futuro, se ha desarrollado un sistema simple de comparación de electrodos para una configuración de fichero dada, y un conjunto de datos. Este proceso, no tiene en cuenta la localidad temporal, así como la procedencia del cluster o la métrica usada. Simplemente realiza una media de los diferentes coeficientes obtenidos para un método sobre un conjunto de datos íntegro.

El objetivo de este tipo de simulaciones sobre el conjunto de electrodos, es el de validar la funcionalidad del proceso, aplicando esta información al clasificador. Esto se incluye en la sección 4.3. Todas las pruebas realizadas en esta sección fueron llevadas a cabo con el método NCD como medida principal de similaridad, haciendo uso del compresor Bzlib. Las razones de elección de este método sobre los demás, aun sin aportar valores superiores a otras técnicas, es la mayor regularidad en su capacidad de detección y la facilidad de modificación del código realizado. Además, al contrario que en las pruebas anteriores, el coeficiente usado para valorar cada electrodo fue el de clase, en lugar del total. La razón fue motivada por la necesidad de obtener aquellos con mayor definición de evento y no aquellas con mejor separación entre evento y la ausencia del mismo.

VI) Búsqueda de sensores

La simulación sobre el conjunto de datos reportó un conjunto de valores para cada electrodo. Los valores obtenidos se muestran en la figura 4.12. En ella se puede observar que existen electrodos con un bajo coeficiente de Silhouette, respecto a otros que llegan a alcanzar el 0.5. Esto puede ser posible, porque el sujeto no presenta gran respuesta en las zonas clasificadas con un bajo coeficiente, o que simplemente esta señal de control no produzca respuestas en esas áreas. En el caso de P300, las zonas con mayor respuesta ante estos estímulos, según la numeración de la competición deberían de ser los electrodos: 34, 11, 49, 51, 53, 56, 62, 60. Aunque en otras competiciones se han desarrollado búsquedas de electrodos con resultados variados [10].

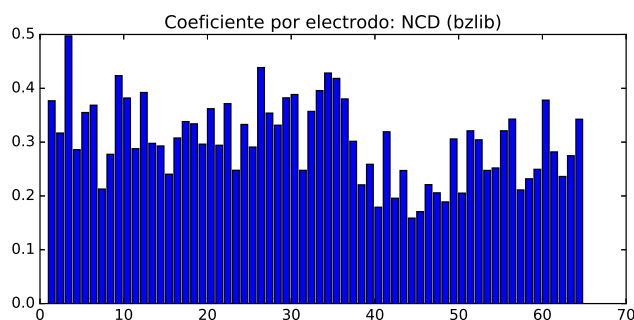


Figura 4.12: Puntuaciones asignadas a cada electrodo mediante NCD con Bzlib

Los resultados obtenidos, por tanto, no siguen el estándar definido para este tipo de problemas, y por desgracia no es posible comparar los resultados de estos por separado con el resto de escogidos, debido a que la selección es completada a los veinte primeros. Y por tanto, con

una diferencia tan grande la comparación no sería realista. No obstante, esto es así debido a la simpleza del clasificador, que quizás podría ser mejorada para hacer uso de este método con un conjunto menor de electrodos.

4.3. Clasificación

Como se adelanta en la sección 2.3, la desconexión persistente en muchos de los métodos o análisis desarrollados con el paradigma a resolver, es un problema común en el contexto de BCI. En este proyecto, con el objetivo de no caer en este error y poder proporcionar una métrica empírica del efecto de algunas de las técnicas aplicadas, se desarrolló un clasificador simple, descrito en la sección 3.1.4.

En esta sección, se muestran los resultados obtenidos de iterar este clasificador sobre el conjunto de datos, descrito anteriormente, y alterando la selección general de canales mediante la información obtenida de las pruebas de puntuación de este. De este modo, dado un clasificador lo suficientemente simple como para tener fallos sobre el conjunto de datos, intentar minimizar este error, mediante una adaptación directa eliminando aquellos electrodos sin información relevante.

4.3.1. Clasificador simple

El clasificador usado en esta prueba, descrito en profundidad en la sección 3.1.4, incluye un paso banda entre 0.5 y 10 Hz, junto con un resampleo a un cuarto de su frecuencia. A continuación, los canales son reconstruidos en nuevas dimensiones a través de PCA, y los N primeros son concatenados como vector resultante de datos para la SVM final. Este sistema presenta una solución sencilla pero útil, dados sus resultados. Al obtener unos coeficientes de sensibilidad superiores al 0.8, es posible tomar como referencia este clasificador con el objetivo de mejorarlo con los métodos usados en este proyecto. De cualquier modo, este coeficiente no representa la clasificación final del sistema, debido a que al tratarse de un sistema por votación, con que los verdaderos positivos sean mayores que el máximo de los falsos positivos correspondientes a un solo estímulo, el sistema se clasificará correctamente.

Los resultados de las pruebas se muestran en la tabla 4.12, separados en tres medidas distintas: La sensibilidad ante el conjunto de entrenamiento, el conjunto test y el total. Del mismo modo, los resultados se encuentran divididos por subconjunto, y, en columnas, por el balanceo de entrenamiento y test. En este último, es apreciable que a mayor conjunto de entrenamiento, más aumenta la sensibilidad del conjunto de test, pero también es necesario tener en cuenta que el conjunto de test se verá mermado en el proceso. No obstante todos los datos mostrados son calculados de forma relativa, e iterados veinticinco veces para hacer la medida más regular.

En la tabla 4.12, se muestra el resultado de la ejecución del clasificador simple sobre los subconjuntos de datos del conjunto definido en 3.4.1. Debido a la necesidad de balancear los datos, debido a que una clase posee 5 veces más datos que la otra (2.1.2) y manteniendo la simpleza del proceso, existe una componente aleatoria en la construcción del conjunto de entrenamiento final. Por ello, en las pruebas realizadas el proceso se repitió veinticinco veces, con el objetivo de obtener una medida regular, y realista, del sistema.

Los resultados obtenidos, muestran que el clasificador tiene una capacidad alta de detectar los eventos de una clase sobre la otra. No obstante, el objetivo final es el de detectar la letra objetivo. Como dato, en el desarrollo del clasificador se obtuvo porcentajes de detección finales demasiado altos, como para hacer uso del clasificador como referencia, por lo que se alteraron parámetros y ajustes para reducir esta tasa y poder intentar mejorarla de manera alternativa.

S. 11				
Trn/Tst	1/5	2/4	3/3	4/2
Train	0.93	0.92	0.916	0.909
Test	0.56	0.629	0.695	0.720
Total	0.726	0.773	0.804	0.812

S. 10			
Trn/Tst	1/4	2/3	3/2
Train	0.9567	0.934	0.918
Test	0.67492	0.684	0.720
Total	0.81456	0.808	0.818

Tabla 4.12: Resultados de la clasificación del conjunto de datos introducido en 3.4.1, para cada sesión.

Trn/Tst	Best				Worst			
	1/5	2/4	3/3	4/2	1/5	2/4	3/3	4/2
Train	0.95538	0.93217	0.92317	0.91766	0.95708	0.9413	0.92092	0.91158
Test	0.57189	0.65519	0.72861	0.76537	0.54549	0.6072	0.66512	0.69581
Total	0.70755	0.79525	0.82235	0.83763	0.67838	0.77077	0.78625	0.79502
Error/575	464	295	100	35	464	370	193	111

Tabla 4.13: Resultados de la clasificación del conjunto de datos introducido en 3.4.1, para cada configuración de electrodos de acuerdo con los obtenidos en la simulación 4.2.4 .

En las pruebas sucesivas se incluye el número de parámetros acertados para cada prueba con objetivo orientativo.

4.3.2. Clasificador modificado

Como primera aproximación a la obtención de una métrica de evaluación del conjunto del sistema, se hizo uso de los datos obtenidos en 4.2.4. De este modo se seleccionaron aquellos veinte primeros electrodos, con mayor calidad de cluster, con el objetivo de minimizar el número de fuentes con componentes ruidosas. Para poder comparar estos resultados, se obtuvieron los veinte con menor coeficiente, para evitar que el ruido producido por estos al conjunto fuese resistido por el clasificador. De este modo, no se obtienen aquellos con componentes más ruidosas, o dicho de otro modo, se sustraen aquellos que contienen la información más relevante.

En la tabla 4.13, se muestran ambos conjuntos de pruebas para los diferentes equilibrios entre entrenamiento y test, para cada conjunto de electrodos. Es fácilmente observable que el ajuste proporcionado por los electrodos seleccionados permiten mejorar en gran medida la capacidad final del clasificador. Verificando por tanto, la capacidad de extracción producida por los métodos aplicados.

5

Conclusiones y trabajo futuro

A lo largo de este documento, se han expuesto distintos problemas inherentes al desarrollo de sistemas BCI. La calidad de los datos, la carencia de generalidad en la fuente, o los riesgos que implican ciertas técnicas de adquisición de señal, son algunos de los ejemplos mencionados. Si bien algunas soluciones BCI aportan resultados positivos, siempre son dependientes de restricciones. Dificultades basadas, en su mayoría, en la ausencia o baja capacidad de adaptación a la fuente de datos. Como alternativa, algunas investigaciones abogan por el uso de metodologías de carácter adaptativo.

En este trabajo de fin de máster se han descrito, implementado y probado distintas métricas de similitud con el objetivo de capturar la estructura de la señal. Esto es posible, debido a que el evento asociado al P300 mantiene una estructura común. Medir como los distintos estímulos pertenecientes a esta clase divergen, proporciona un valor de los cambios en la estructura temporal de la señal. Aquellos que mantengan un comportamiento parecido representarán un periodo estacionario, mientras que aquellos que difieran, permitirán identificar un cambio. De este modo, es posible graduar la estructura temporal del evento para un mismo sujeto. O en otras palabras, medir la deriva intra-sujeto. Por otro lado, es razonable pensar que algunos conjuntos mantengan estructuras más similares que otros. Es decir, que mantengan una cohesión mayor entre sus eventos. De este modo y teniendo en cuenta que un mismo usuario proporciona varias fuentes de información, es posible calificar cual de ellas representa de forma más precisa la estructura de la información. Lo cual, proporciona una métrica particular del sujeto en el tiempo. O dicho de otro modo, una adaptación intersujeto.

La intención de este proyecto, era revisar diferentes técnicas de análisis y tratamiento de información con la finalidad de obtener una base de datos para el desarrollo de un futuro sistema de clasificación adaptativa. No obstante, además de completar este conjunto con el resto de objetivos propuestos, se desarrolló un conjunto de sistemas que permitieron identificar características individuales del sujeto en el tiempo. Los resultados obtenidos en las pruebas realizadas, verifican la capacidad de los métodos usados para producir métricas de adaptación. Entendiendo la adaptación en este contexto, como la capacidad de medir propiedades instantáneas y particulares de cada sujeto. De los resultados obtenidos en este proyecto se pueden extraer varias conclusiones.

En primer lugar, las simulaciones realizadas sobre la construcción de los objetos evidencian la capacidad de captura de la información del sistema desarrollado. Esto se puede observar en la mejora de los coeficientes obtenidos a medida que se agrega información al sistema (Representado

en los mapas de color). Útil para capturar la deriva de la estacionaridad local, parcialmente detectada en la caída general de los coeficientes obtenidos para distintos subconjuntos para algunas técnicas de evolución más regular. Este es el caso de NCD, que capta la estructura de forma más progresiva y por tanto más adaptable. Un caso que podría verse beneficiado de este hecho, es la preparación de una toma de datos. Debido a que NCD resulta más eficiente para capturar la información, podría reducirse la cantidad de estímulos del sistema.

Y por otro lado, la mejora resultante de la selección de electrodos, prueba su competencia como complemento a un clasificador real. Esto es parcialmente observable en las pruebas realizadas con el clasificador simple, aunque no completamente aplicable. El hecho de tratarse de un clasificador simple, hace que quizás la información que con este método se evita, calificándola como ruido, en otros sistemas pueda servir de ayuda. Por lo que quizás su aplicación no sea del todo relevante de cara a un problema final, pero si interesante desde el punto de vista de la información que discrimina.

Por todo ello, y como ya se introdujo al inicio de este documento, este proyecto tiene entre sus objetivos el de capacitar de base y formación a un futuro proyecto de tesis, desarrollado en el entorno de la adaptación a conjuntos de datos en el contexto de estudio de la estacionaridad local, y demás características de la información capturables. Como primera aproximación, se podría refinar las capacidades de los distintos métodos desarrollados, ampliando la capacidad de las búsquedas paramétricas. Por ejemplo, agregando definición a estas, o variables no iteradas en este proyecto. Aunque también resultaría interesante, agregar más métodos y conjuntos de datos, para validar las capacidades de estos, y refinar los métodos usados en función de su capacidad.

A modo de transferencia de conocimiento, los resultados obtenidos en este proyecto así como las conclusiones y métodos implementados, se recogerán en un artículo resumen con el propósito de ser publicado. La idea fundamental, basada en la mejora ocasionada por el uso de métodos basados en adaptación, se recogerá en este documento, así como un enlace al código fuente y las recetas que permitan replicar los resultados obtenidos.

Finalmente este trabajo, aunque introductorio, permite comprender la naturaleza del problema, así como las dificultades añadidas. Paralelamente, proporciona una serie de propuestas iniciales, conjunto con resultados de las soluciones implementadas, y sus capacidades iniciales en este contexto.

Glosario de acrónimos

- **EEG**: Electroencefalograma
- **MEG**: Magnetoencefalograma
- **fMRI**: Imagen por Resonancia Magnética funcional
- **ECoG**: Electrocorticograma
- **P300**: Potencial 300ms
- **SCP**: Potenciales Corticales Lentos
- **VEP**: Potencial Visual Evocado
- **SSVEP**: Steady State VEP
- **TEVP**: VEP Transitorio
- **ERP**: Potenciales Relacionados a Evento
- **BCI**: Interfaces Cerebro Computador
- **BMI**: Interfaces Cerebro Máquina
- **UI**: Interfaz de Usuario
- **GUI**: Interfaz de Usuario Gráfica
- **PCA**: Análisis de Componentes Principales
- **ICA**: Análisis de Componentes Independientes
- **DWT**: Transformada Wavelet Discreta
- **MI**: Información Mutua
- **NCD**: Distancia Normalizada por Compresor
- **SL**: Synchronization Likelihood
- **LDA**: Análisis de Discriminante Lineal
- **SVM**: Máquina de Soporte Vectorial

Bibliografía

- [1] Setare Amiri, Ahmed Rabbi, Leila Azinfar, and Reza Fazel-Rezai. A Review of P300, SSVEP, and Hybrid P300/SSVEP Brain- Computer Interface Systems. In Reza Fazel-Rezai, editor, *Brain-Computer Interface Systems - Recent Progress and Future Prospects*. InTech, June 2013.
- [2] T. W. Picton. The P300 wave of the human event-related potential. *Journal of Clinical Neurophysiology: Official Publication of the American Electroencephalographic Society*, 9(4):456–479, October 1992.
- [3] Rik van Dinteren, Martijn Arns, Marijtje L. A. Jongsma, and Roy P. C. Kessels. P300 Development across the Lifespan: A Systematic Review and Meta-Analysis. *PLoS ONE*, 9(2):e87347, February 2014.
- [4] C. E. van Beijsterveldt and D. I. Boomsma. Genetics of the human electroencephalogram (EEG) and event-related brain potentials (ERPs): a review. *Human Genetics*, 94(4):319–330, October 1994.
- [5] Christoph Klein, Katharina Wendling, Paul Huettner, Hans Ruder, and Martin Peper. Intra-subject variability in attention-deficit hyperactivity disorder. *Biological Psychiatry*, 60(10):1088–1097, November 2006.
- [6] Y. Tran, A. Craig, and P. McIsaac. Extraversion-introversion and 8-13 Hz waves in frontal cortical regions. *Personality and Individual Differences*, 2(30):205–215, 2001.
- [7] Jacobo Fernandez-Vargas, Hanns Uwe Pfaff, Francisco B. Rodriguez, and Pablo Varona. Assisted closed-loop optimization of SSVEP-BCI efficiency. *Frontiers in Neural Circuits*, 7:27, 2013.
- [8] Pradeep Shenoy, Matthias Krauledat, Benjamin Blankertz, Rajesh P. N. Rao, and Klaus-Robert Mäijller. Towards adaptive classification for BCI. *Journal of Neural Engineering*, 3(1):R13–23, March 2006.
- [9] Neng Xu, Xiaorong Gao, Bo Hong, Xiaobo Miao, Shangkai Gao, and Fusheng Yang. BCI Competition 2003–Data set IIb: enhancing P300 wave detection using ICA-based subspace projections for BCI applications. *IEEE transactions on bio-medical engineering*, 51(6):1067–1072, June 2004.
- [10] A. Rakotomamonjy and V. Guigue. BCI Competition III: Dataset II- Ensemble of SVMs for BCI P300 Speller. *IEEE Transactions on Biomedical Engineering*, 55(3):1147–1154, March 2008.
- [11] Setare Amiri, Reza Fazel-Rezai, Vahid Asadpour, Setare Amiri, Reza Fazel-Rezai, and Vahid Asadpour. A Review of Hybrid Brain-Computer Interface Systems, A Review of Hybrid Brain-Computer Interface Systems. *Advances in Human-Computer Interaction, Advances in Human-Computer Interaction*, 2013, 2013:e187024, February 2013.

- [12] Suresh D. Muthukumaraswamy. High-frequency brain activity and muscle artifacts in MEG/EEG: a review and recommendations. *Frontiers in Human Neuroscience*, 7, April 2013.
- [13] Christoph Guger, Gunther Krausz, Brendan Z. Allison, and Guenter Edlinger. Comparison of dry and gel based electrodes for P300 brain-computer interfaces. *Neuroprosthetics*, 6:60, 2012.
- [14] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain Computer Interfaces, a Review. *Sensors (Basel, Switzerland)*, 12(2):1211–1279, January 2012.
- [15] T. D. Lagerlund, F. W. Sharbrough, C. R. Jack, B. J. Erickson, D. C. Strelow, K. M. Cicora, and N. E. Busacker. Determination of 10-20 system electrode locations using magnetic resonance image scanning with markers. *Electroencephalography and Clinical Neurophysiology*, 86(1):7–14, January 1993.
- [16] Min Hye Chang, Jeong Su Lee, Jeong Heo, and Kwang Suk Park. Eliciting dual-frequency SSVEP using a hybrid SSVEP-P300 BCI. *Journal of Neuroscience Methods*, 258:104–113, 2016.
- [17] UmmAijhan Işoğlu-alkağ, Karina Kedzior, Sacit KaramAjrşel, and Numan Ermutlu. Event-related potentials during auditory oddball, and combined auditory oddball-visual paradigms. *The International Journal of Neuroscience*, 117(4):487–506, April 2007.
- [18] A Tactile P300 Brain-Computer Interface.
- [19] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain-computer interfaces. *Journal of Neural Engineering*, 4(2):R1, 2007.
- [20] Reza Fazel-Rezai, Brendan Z. Allison, Christoph Guger, Eric W. Sellers, Sonja C. Kleih, and Andrea Kübler. P300 brain computer interface: current challenges and emerging trends. *Frontiers in Neuroengineering*, 5:14, 2012.
- [21] Han Yuan and Bin He. Brain-computer interfaces using sensorimotor rhythms: current state and future perspectives. *IEEE transactions on bio-medical engineering*, 61(5):1425–1435, May 2014.
- [22] Nicolas Brodu, Fabien Lotte, and Anatole Lécuyer. Exploring two novel features for EEG-based brain-computer interfaces: Multifractal cumulants and predictive complexity. *Neurocomputing*, 79:87–94, March 2012.
- [23] Francesca Miraglia, Fabrizio Vecchio, Placido Bramanti, and Paolo Maria Rossini. EEG characteristics in eyes-open versus eyes-closed conditions: Small-world network architecture in healthy aging and age-related brain degeneration. *Clinical Neurophysiology*, 127(2):1261–1268, February 2016.
- [24] Pradeep Shenoy, Matthias Krauledat, Benjamin Blankertz, Rajesh P. N. Rao, and Klaus-Robert Müller. Towards adaptive classification for BCI. *Journal of Neural Engineering*, 3(1):R13, 2006.
- [25] Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic Classification of Artifactual ICA-Components for Artifact Removal in EEG Signals. *Behavioral and Brain Functions : BBF*, 7:30, August 2011.

- [26] Abdulhamit Subasi and M. Ismail Gursoy. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, 37(12):8659–8666, December 2010.
- [27] Cosma Rohilla Shalizi and Kristina Lisa Shalizi. Blind Construction of Optimal Nonlinear Recursive Predictors for Discrete Sequences. *arXiv:cs/0406011*, June 2004. arXiv: cs/0406011.
- [28] Nicolas Brodu. Reconstruction of Epsilon-Machines in Predictive Frameworks and Decisional States. *arXiv:0902.0600 [stat]*, February 2009. arXiv: 0902.0600.
- [29] James P. Crutchfield and Karl Young. Inferring statistical complexity. *Physical Review Letters*, 63(2):105–108, July 1989.
- [30] Stéphane Jaffard Patrice Abry. Revisiting scaling, multifractal, and multiplicative cascades with the wavelet leader lens. *Proceedings of SPIE - The International Society for Optical Engineering*, 5607, 2004.
- [31] N. Brodu. Multifractal feature vectors for Brain-Computer interfaces. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 2883–2890, June 2008.
- [32] A. Granados, K. Koroutchev, and F. d B. Rodríguez. Discovering Data Set Nature through Algorithmic Clustering Based on String Compression. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):699–711, March 2015.
- [33] Rudi Cilibrasi and Paul Vitányi. Clustering by compression. *arXiv:cs/0312044*, December 2003. arXiv: cs/0312044.
- [34] Kristina Lisa Klinkner, Cosma Rohilla Shalizi, and Marcelo F. Camperi. Measuring Shared Information and Coordinated Activity in Neuronal Networks. *arXiv:q-bio/0506009*, June 2005. arXiv: q-bio/0506009.
- [35] C. J. Stam and B. W. van Dijk. Synchronization likelihood: an unbiased measure of generalized synchronization in multivariate data sets. *Physica D: Nonlinear Phenomena*, 163(3–4):236–251, March 2002.
- [36] Rudi Cilibrasi and Paul M. B. Vitányi. A New Quartet Tree Heuristic for Hierarchical Clustering. *arXiv:cs/0606048*, June 2006. arXiv: cs/0606048.
- [37] R. Cilibrasi and P. M. B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, April 2005.
- [38] Jiaxin Ma, Yu Zhang, Andrzej Cichocki, and Fumitoshi Matsuno. A Novel EOG/EEG Hybrid Human–Machine Interface Adopting Eye Movements and ERPs: Application to Robot Control. *IEEE Transactions on Biomedical Engineering*, 62(3):876–889, March 2015.
- [39] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987.
- [40] M. Kaper, P. Meinicke, U. Grossekhoefer, T. Lingner, and H. Ritter. BCI competition 2003-data set IIB: support vector machines for the P300 speller paradigm. *IEEE Transactions on Biomedical Engineering*, 51(6):1073–1076, June 2004.

- [41] B. Blankertz, K. R. Muller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlogl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schroder, and N. Birbaumer. The BCI competition 2003: progress and perspectives in detection and discrimination of EEG single trials. *IEEE Transactions on Biomedical Engineering*, 51(6):1044–1051, June 2004.



Resultados de las simulaciones

El proposito de este anexo es incluir los resultados de las diferentes simulaciones, para cualquier consulta referedia a los mismos, durante la lectura de este documento.

A.1. Simulaciones sobre parámetros de estructura

En la sección 4.2.3, se realizan diferentes simulaciones de las cuales se muestran los resultados de un subconjunto de los datos totales. En esta sección se incluyen los resultados parcialmente completos, al incluir dos (total y clase 1), de los tres mapas de color resultantes de la simulación de cada subconjunto. Adicionalmente se incluyen los dendogramas obtenidos para algunos puntos individuales de la simulación, a modo de referencia.

A.1.1. Bzlib: conjunto A, subconjuntos del 1 al 5

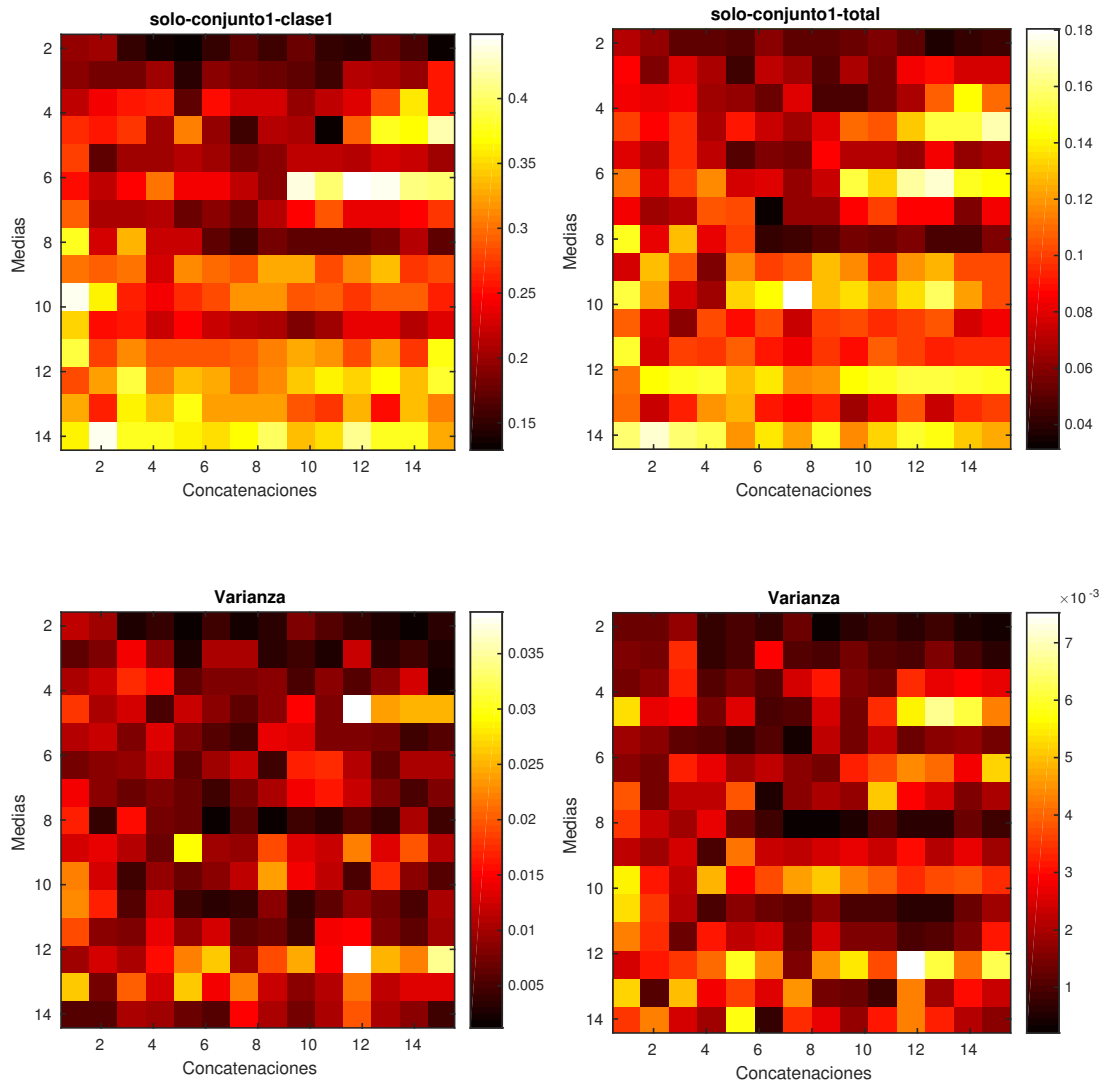


Figura A.1: Mapa de color del subconjunto 1 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

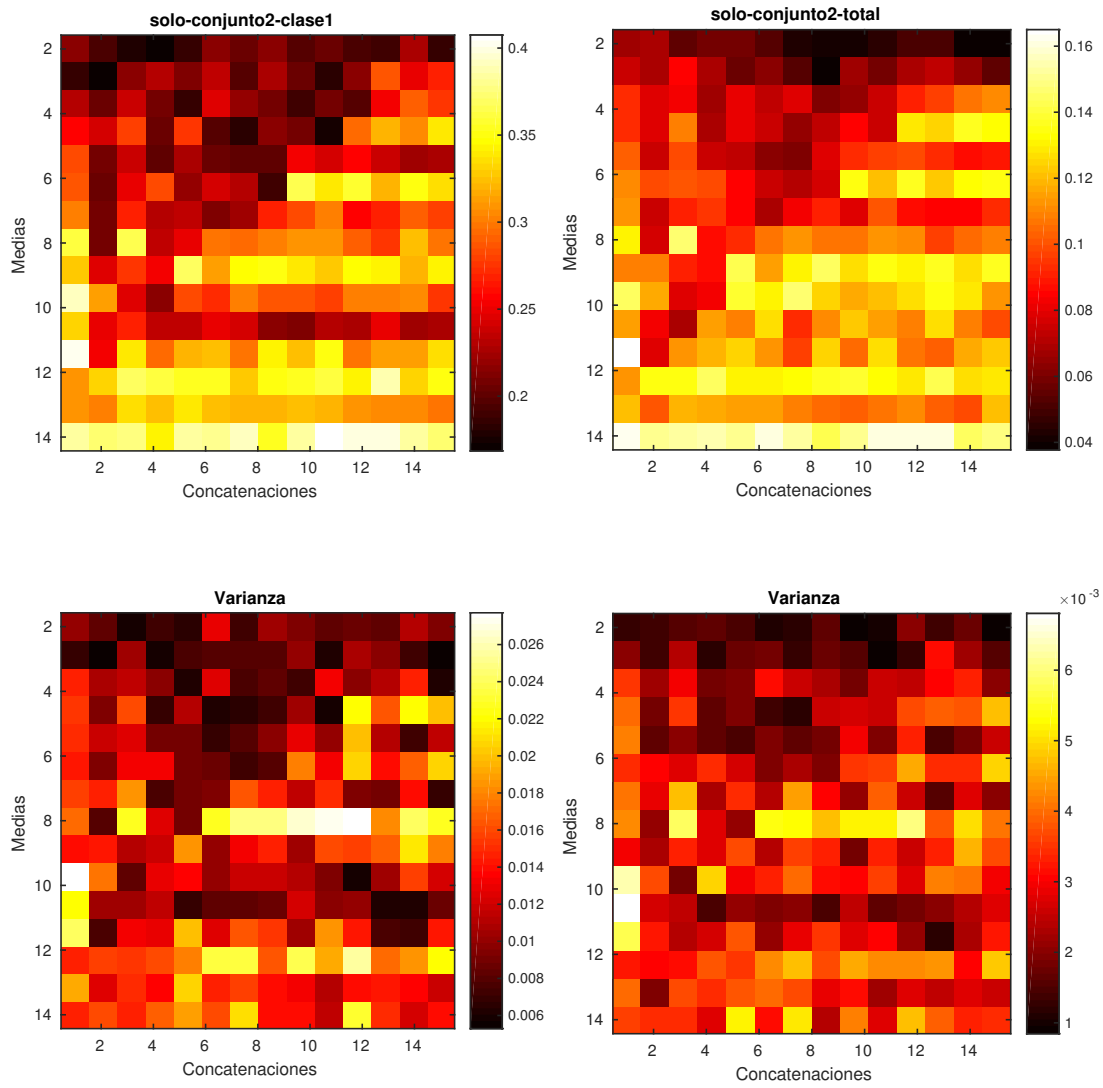


Figura A.2: Mapa de color del subconjunto 2 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

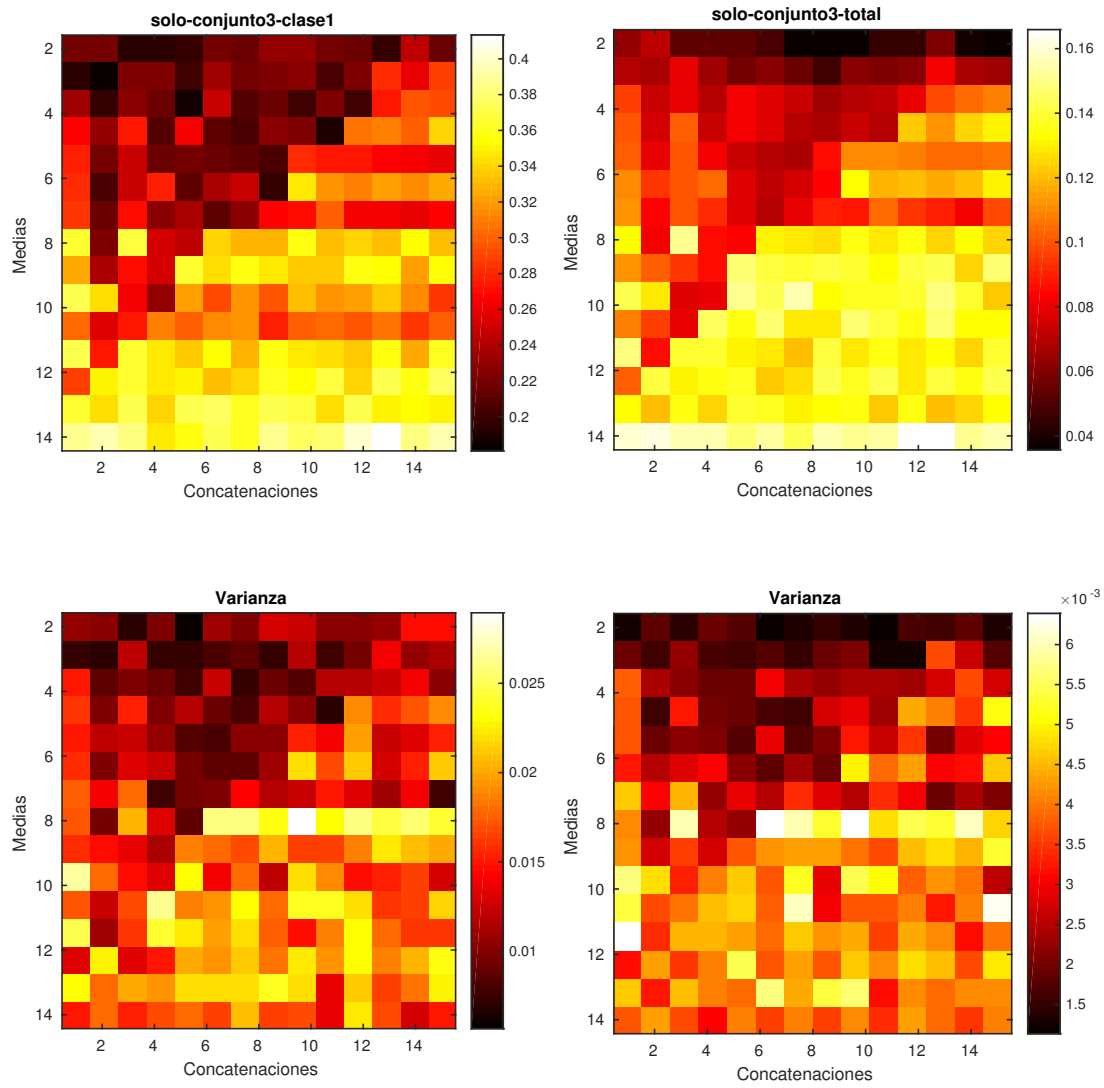


Figura A.3: Mapa de color del subconjunto 3 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

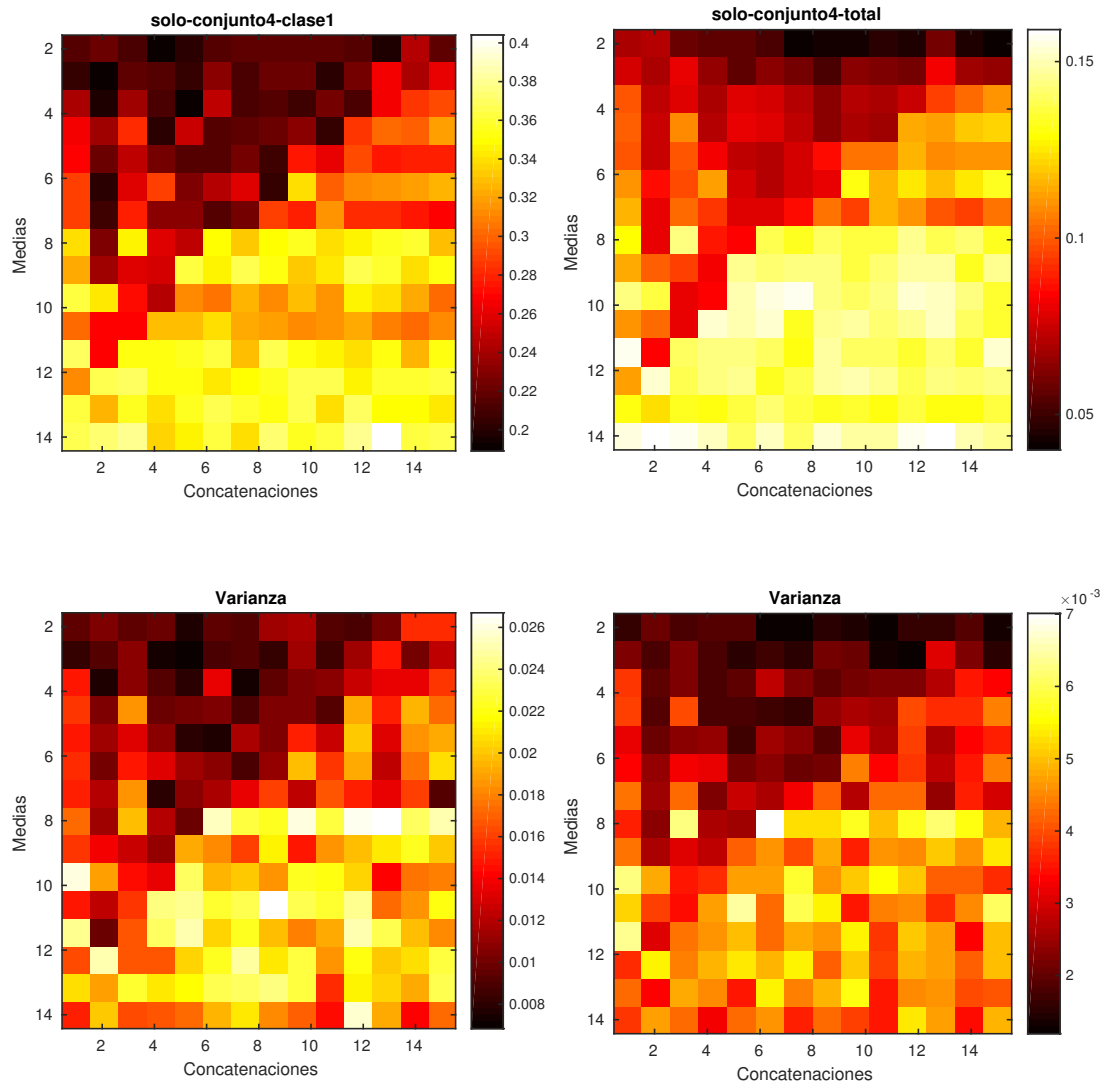


Figura A.4: Mapa de color del subconjunto 4 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

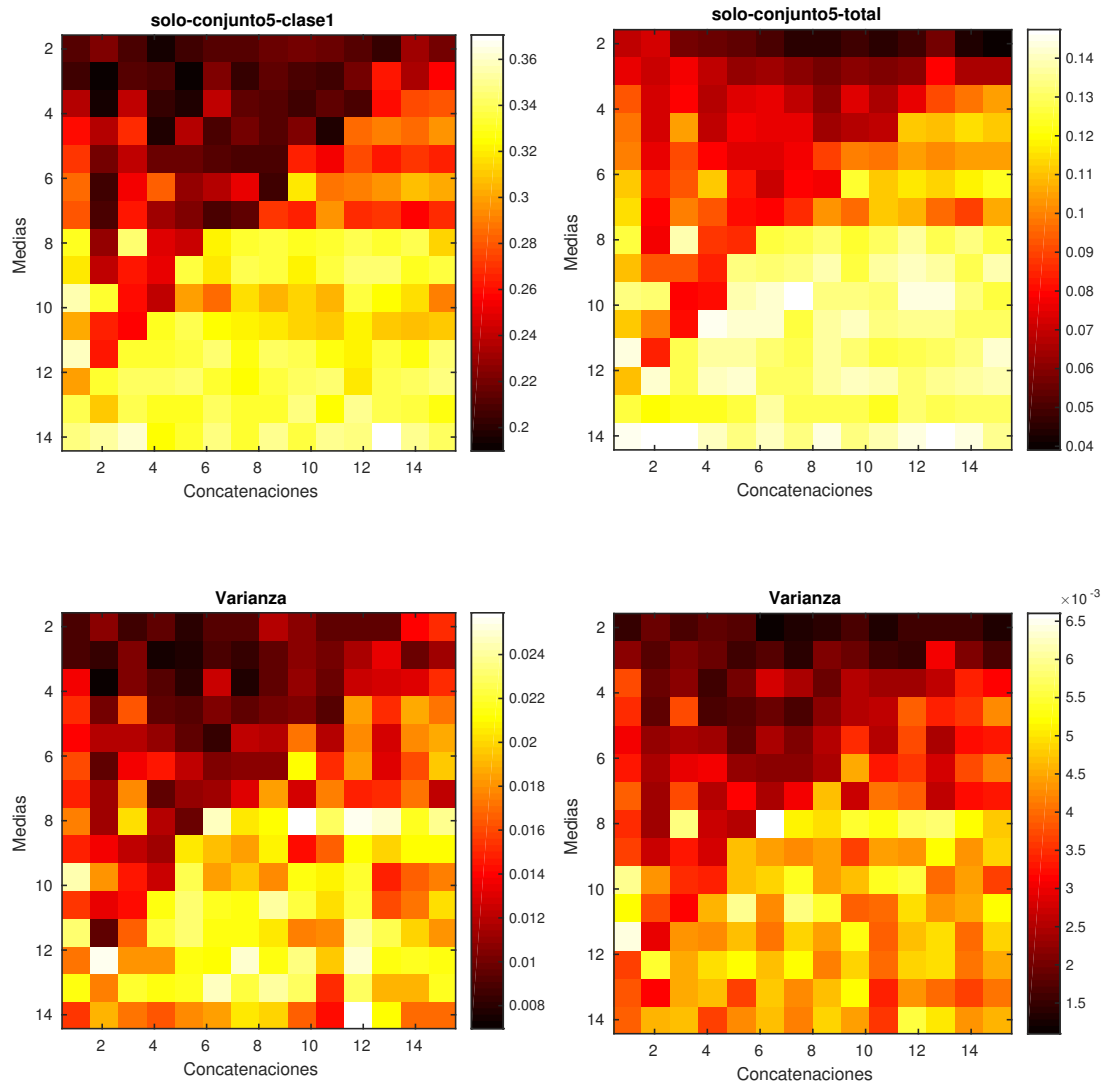


Figura A.5: Mapa de color del subconjunto 5 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.2. Bzlib: conjunto A, subconjuntos del 1 al 5 por invasión

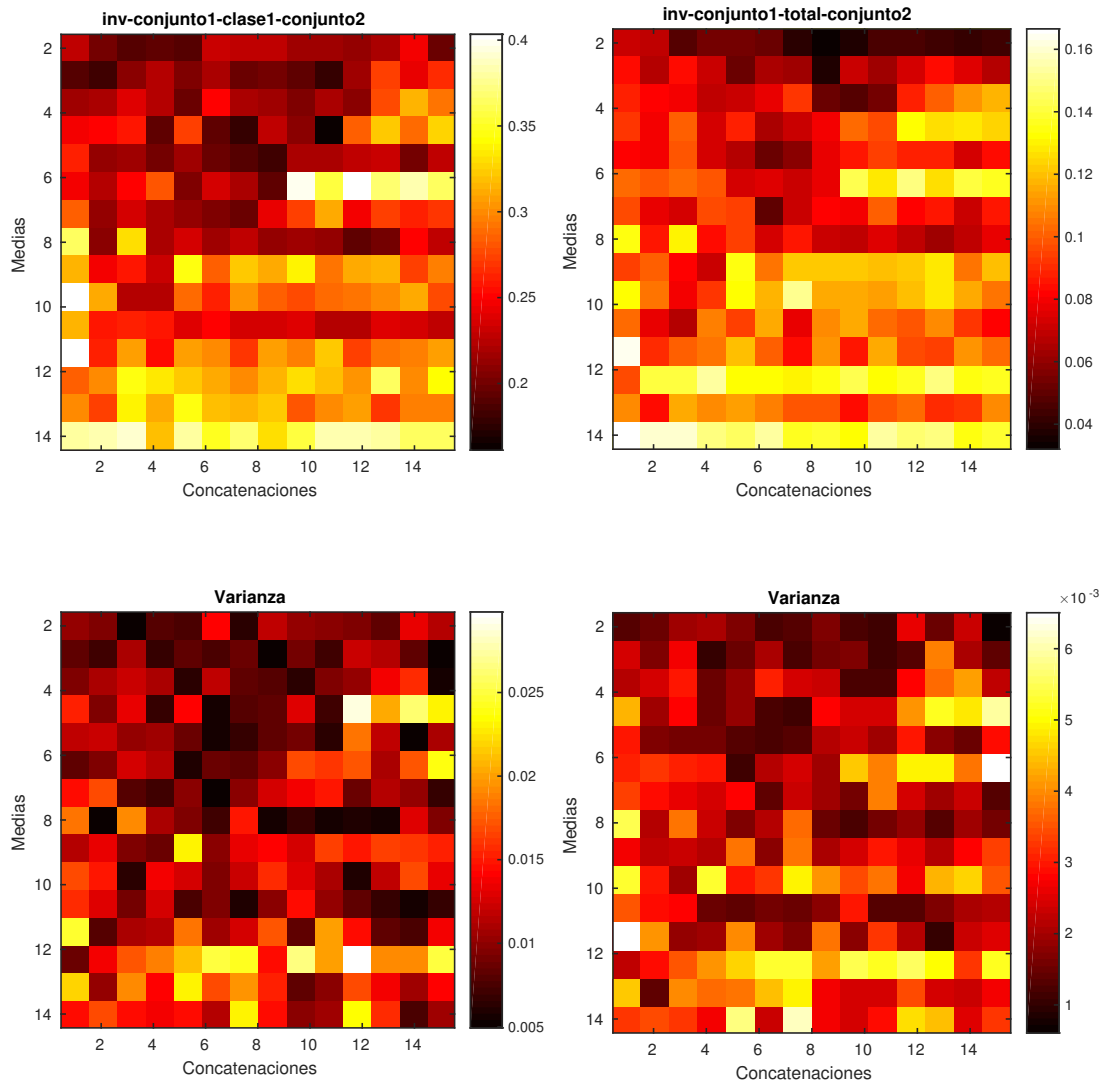


Figura A.6: Mapa de color del subconjunto 1 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

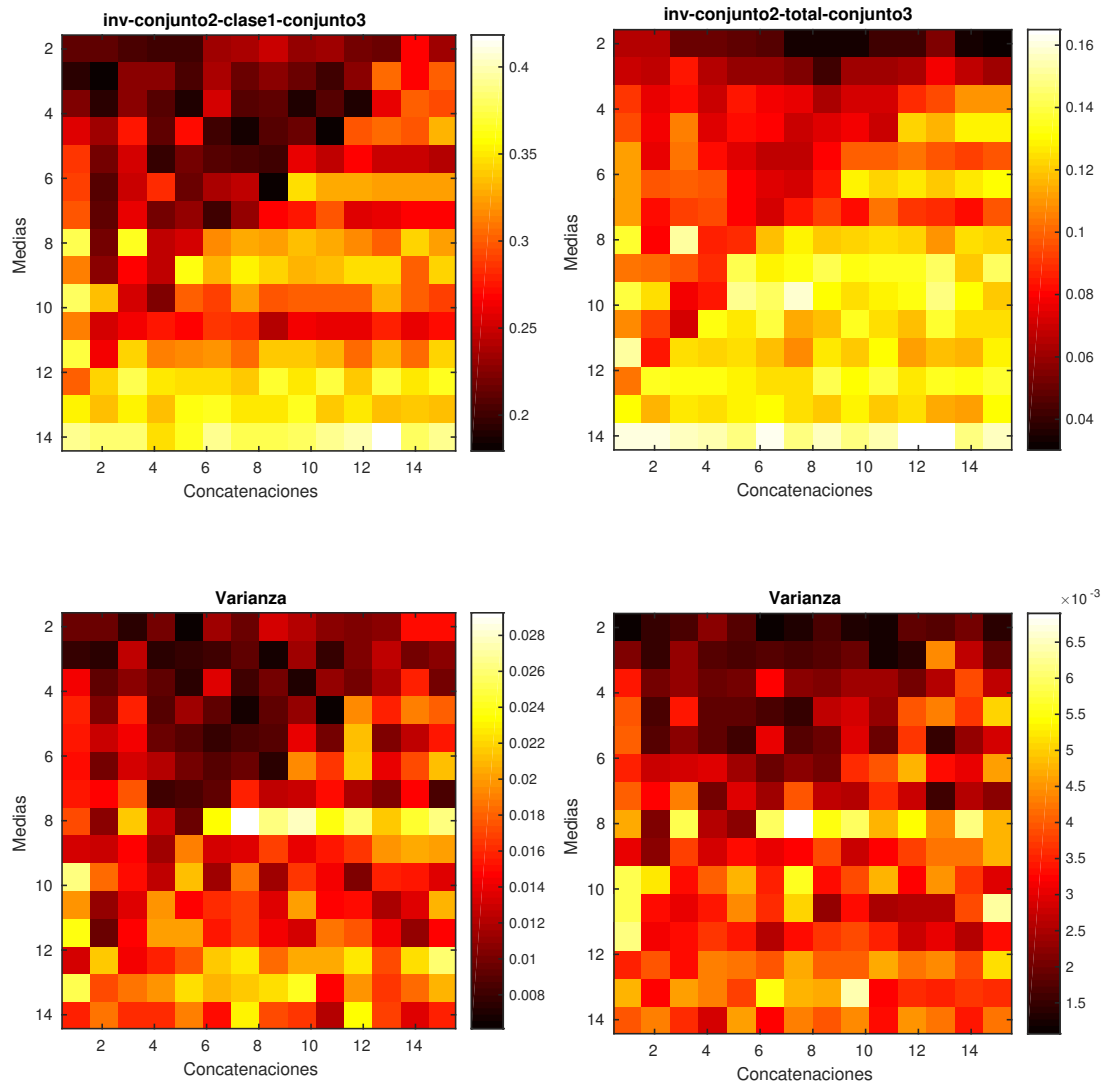


Figura A.7: Mapa de color del subconjunto 2 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

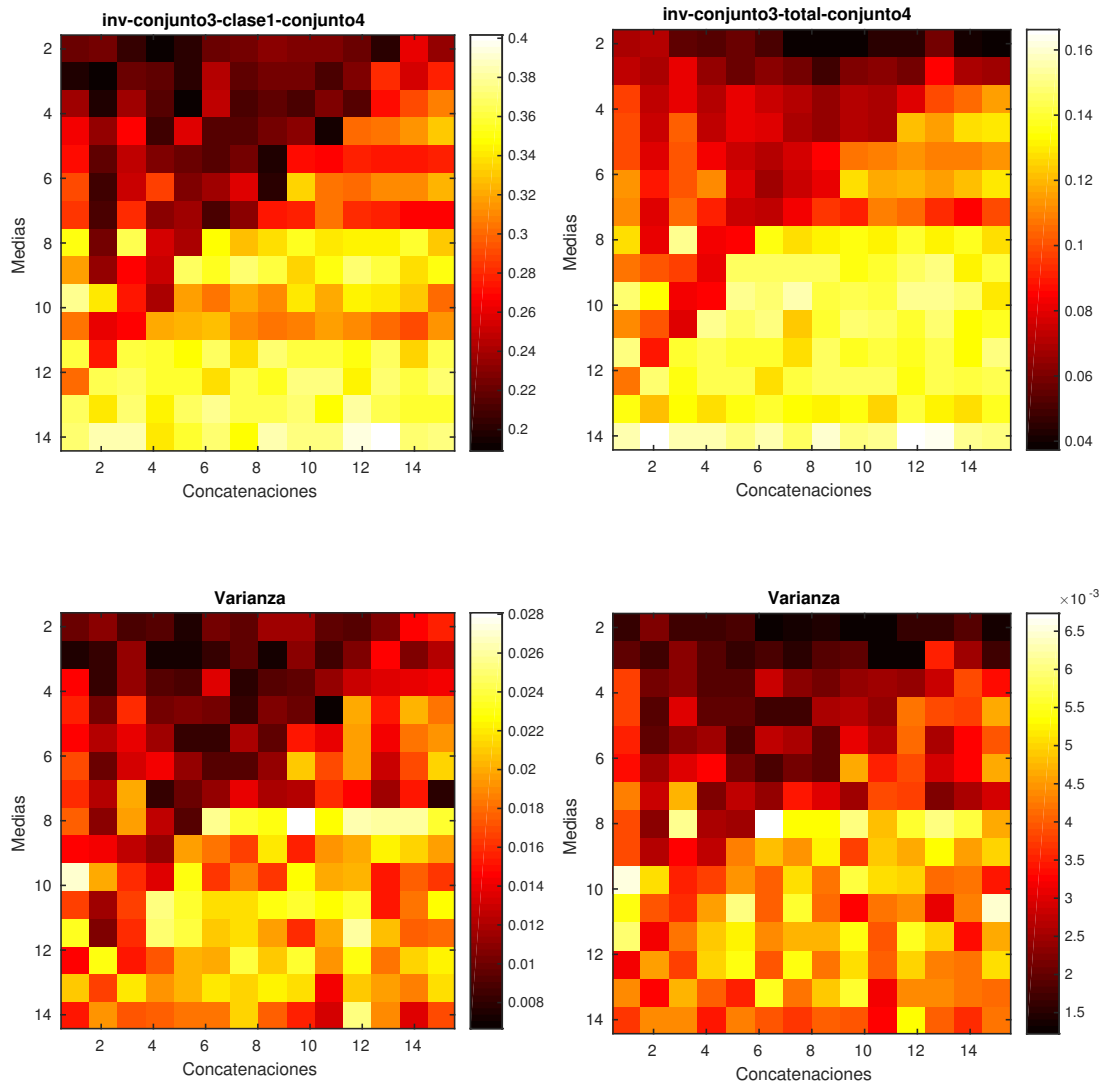


Figura A.8: Mapa de color del subconjunto 3 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

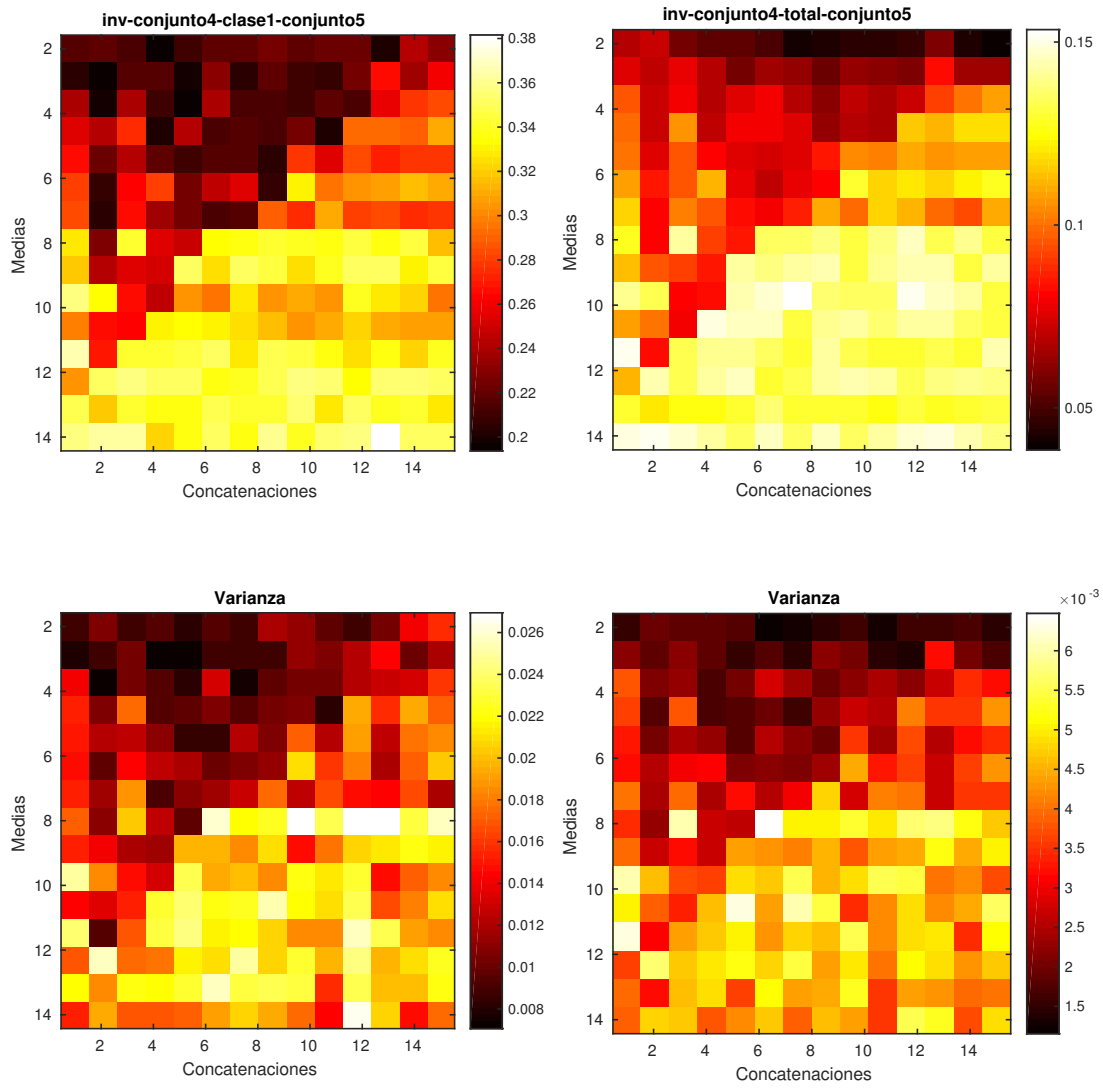


Figura A.9: Mapa de color del subconjunto 4 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

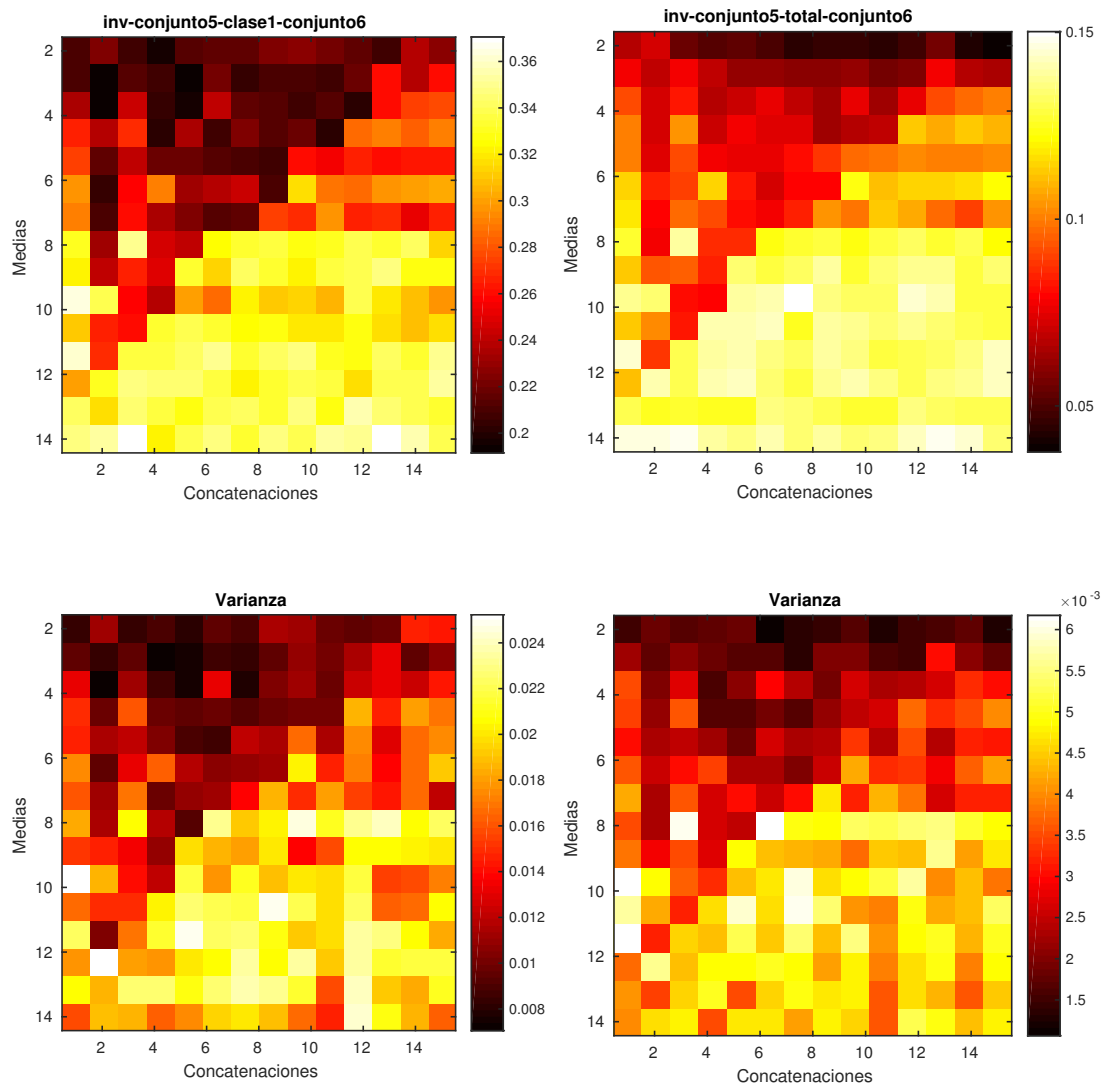


Figura A.10: Mapa de color del subconjunto 5 del método bzlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.3. Zlib: conjunto A, subconjuntos del 1 al 5

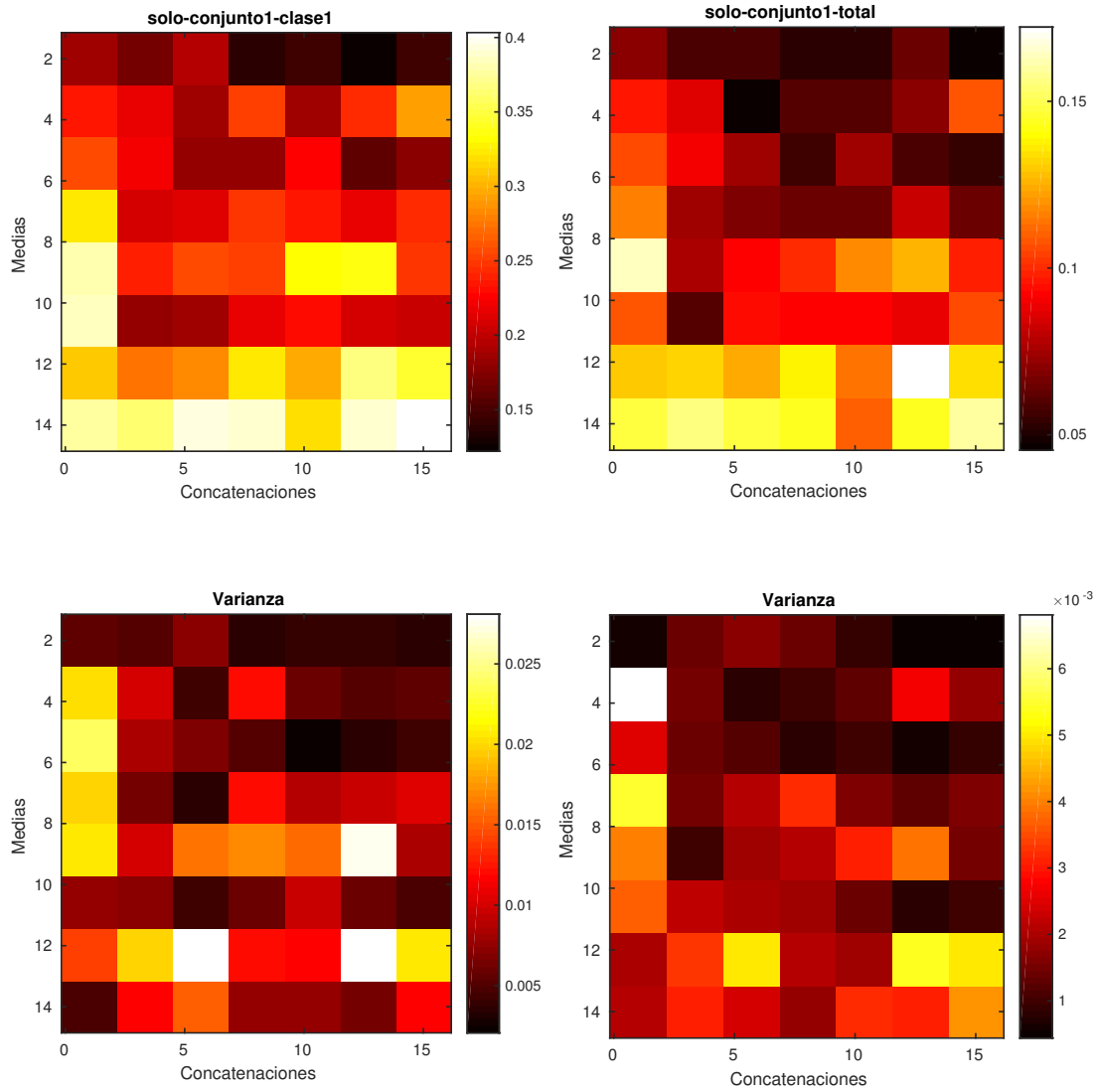


Figura A.11: Mapa de color del subconjunto 1 del método zlib. El coeficiente representado a la izquierda representa la clase del ERP P300, y el de la derecha el total.

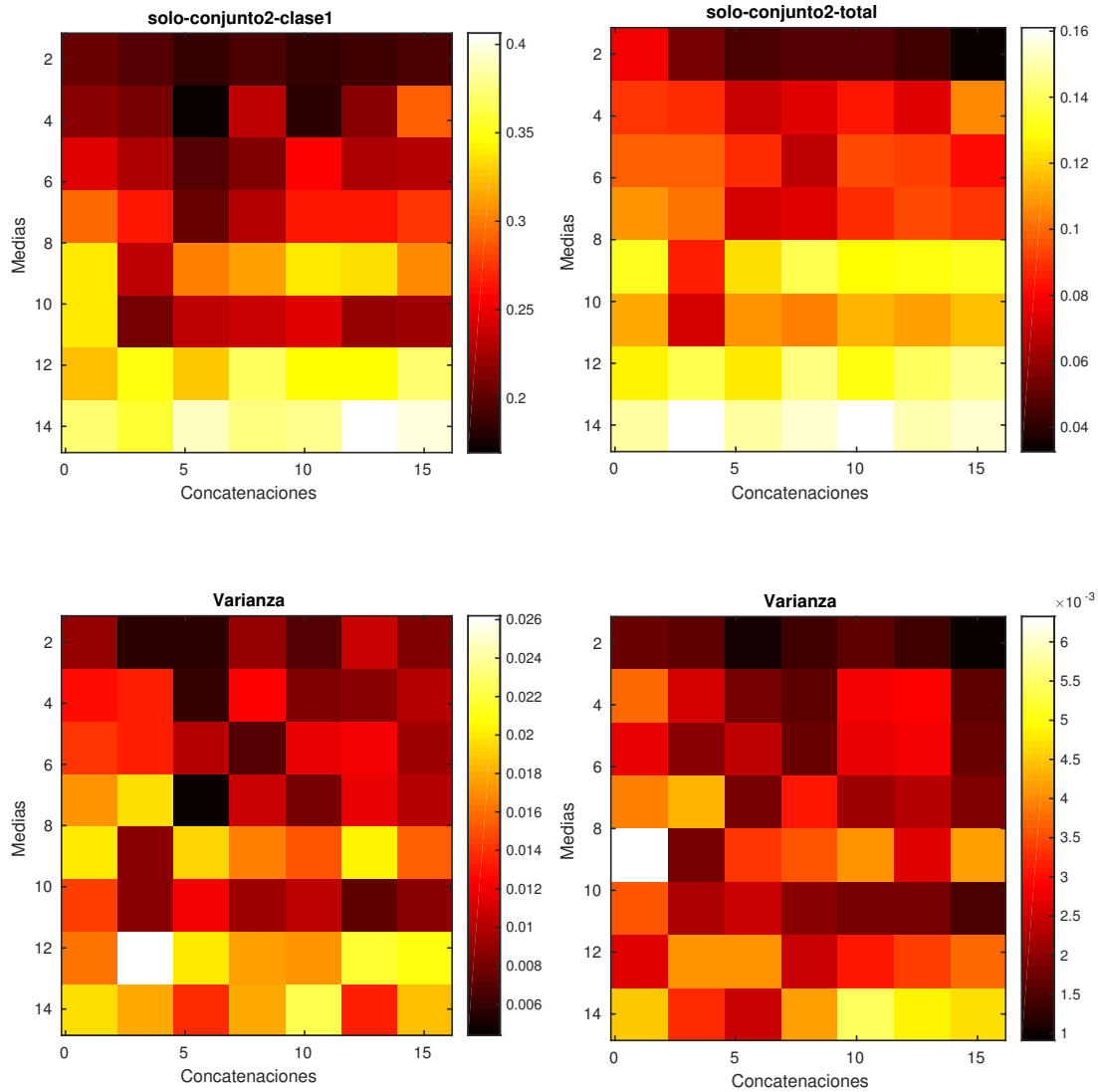


Figura A.12: Mapa de color del subconjunto 2 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

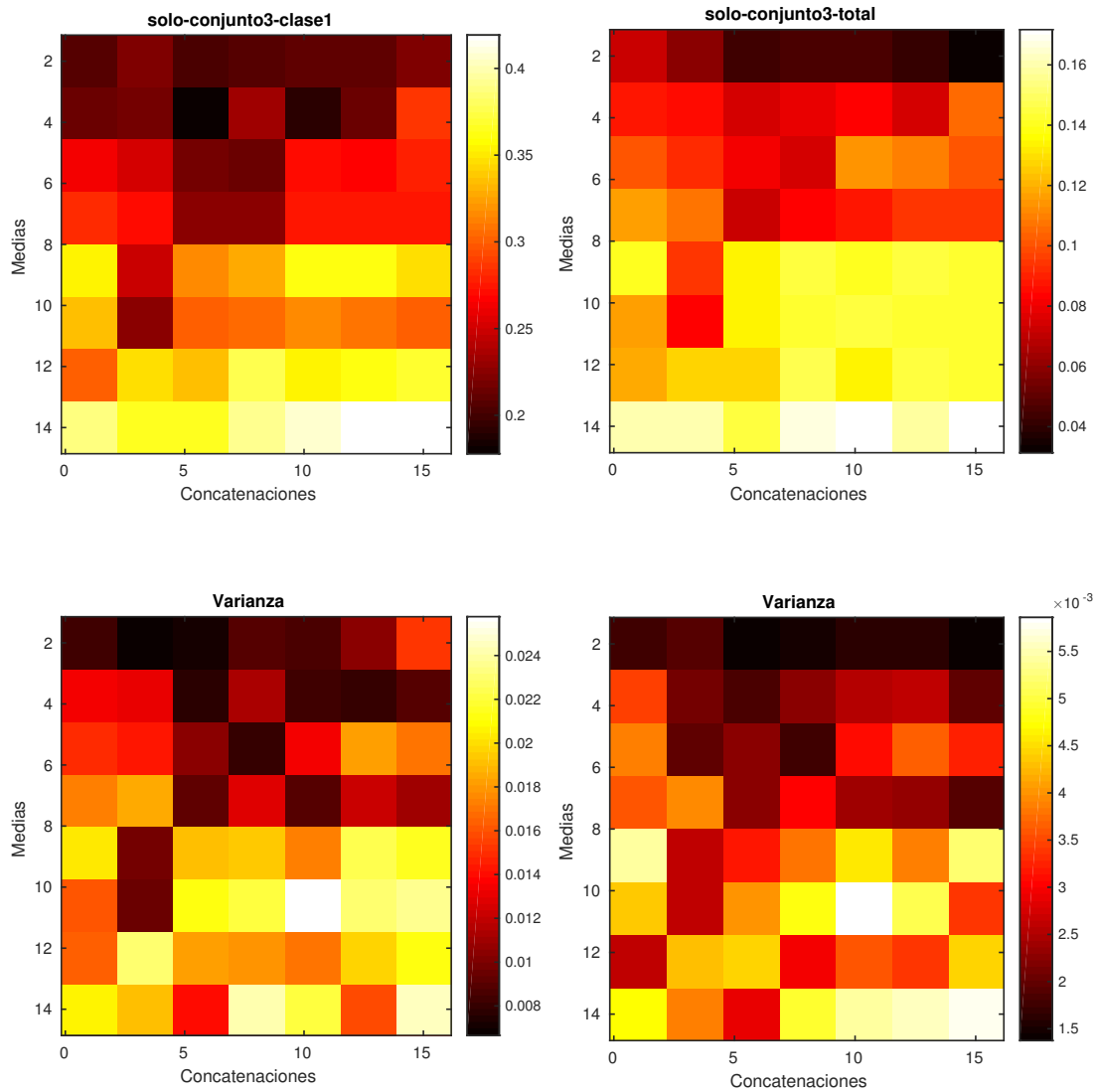


Figura A.13: Mapa de color del subconjunto 3 del método zlib. El coeficiente representado a la izquierda representa la clase del ERP P300, y el de la derecha el total.

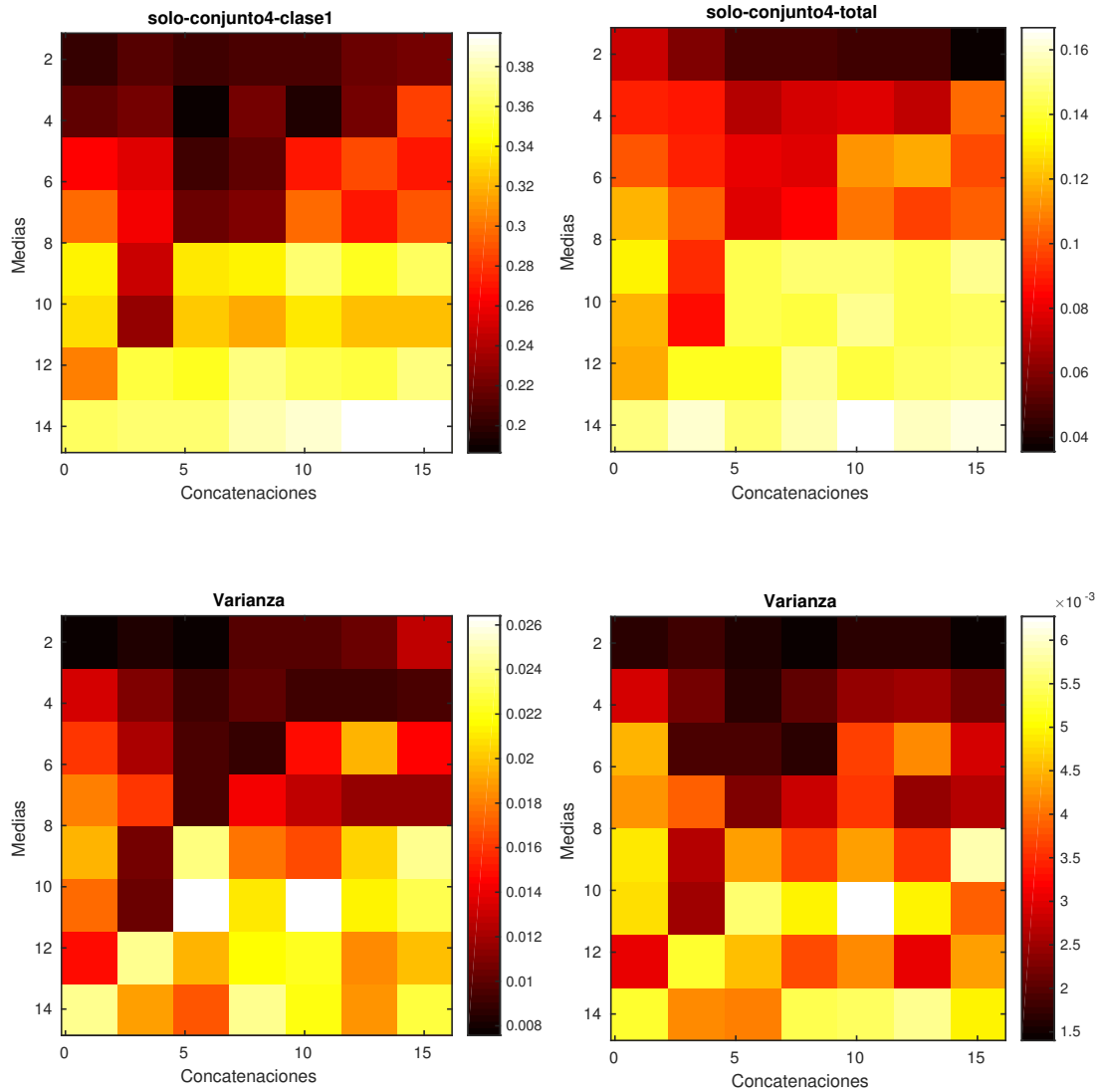


Figura A.14: Mapa de color del subconjunto 4 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

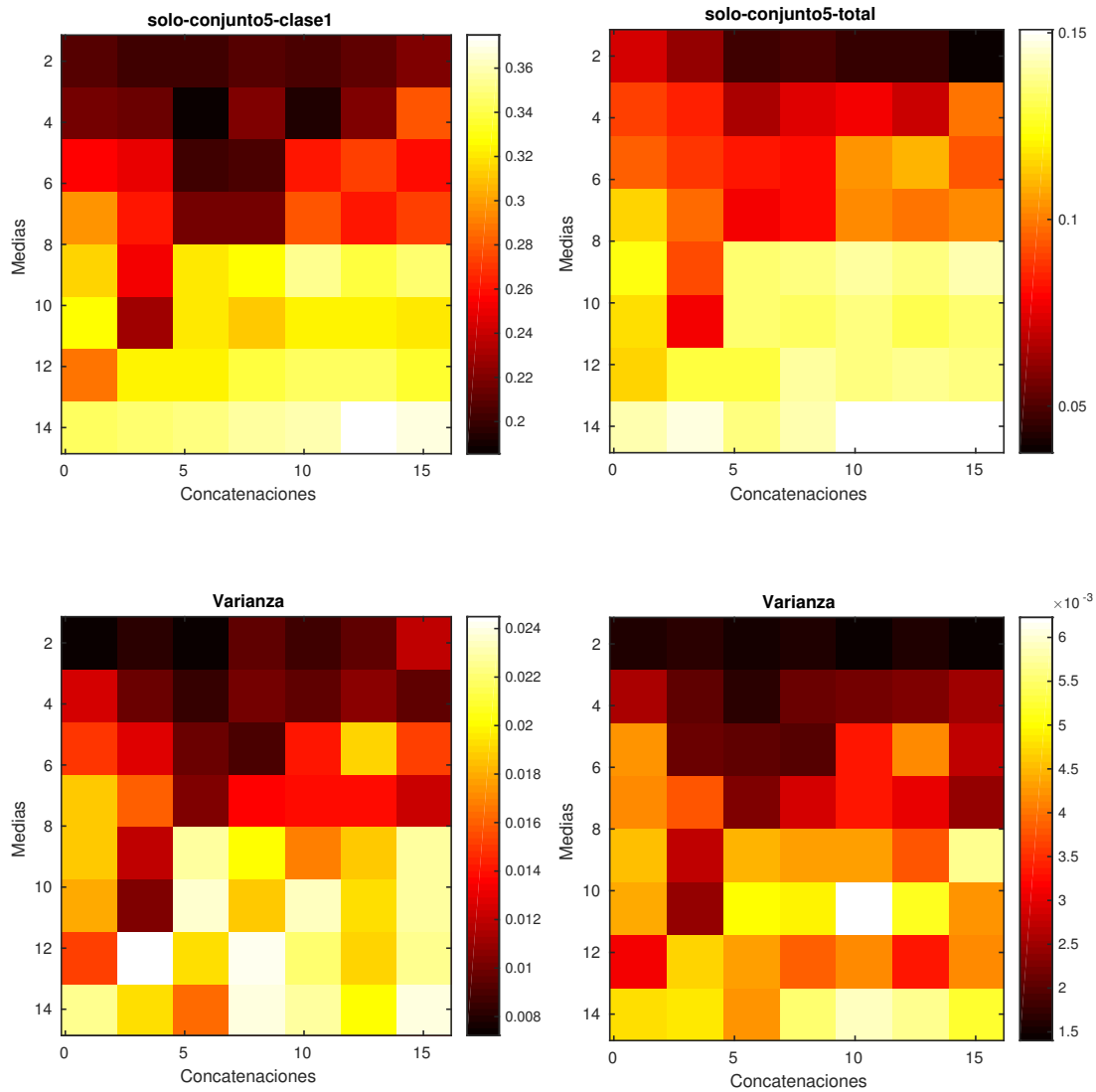


Figura A.15: Mapa de color del subconjunto 5 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.4. Zlib: conjunto A, subconjuntos del 1 al 5 por invasión

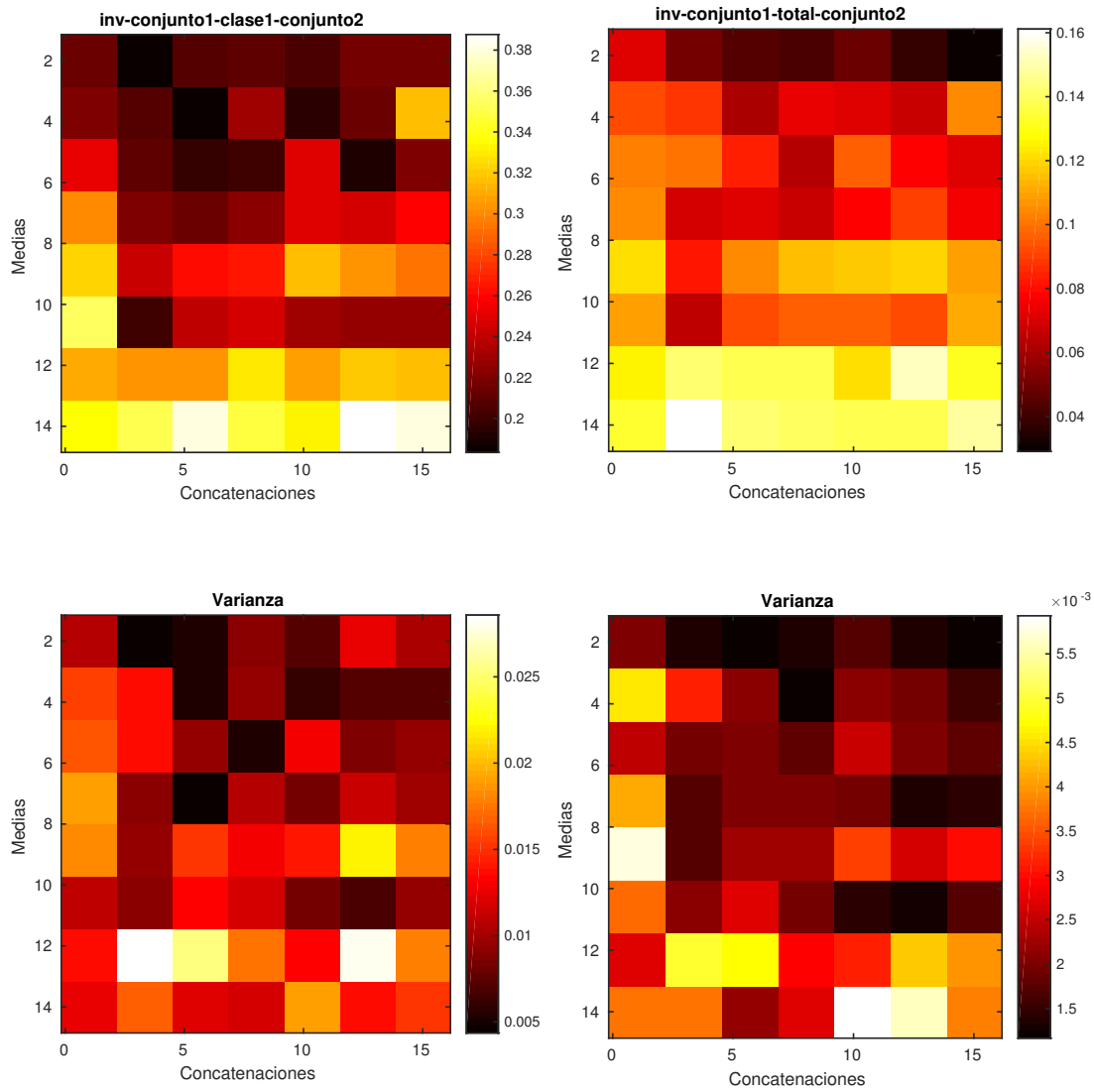


Figura A.16: Mapa de color del subconjunto 1 del método zlib. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.

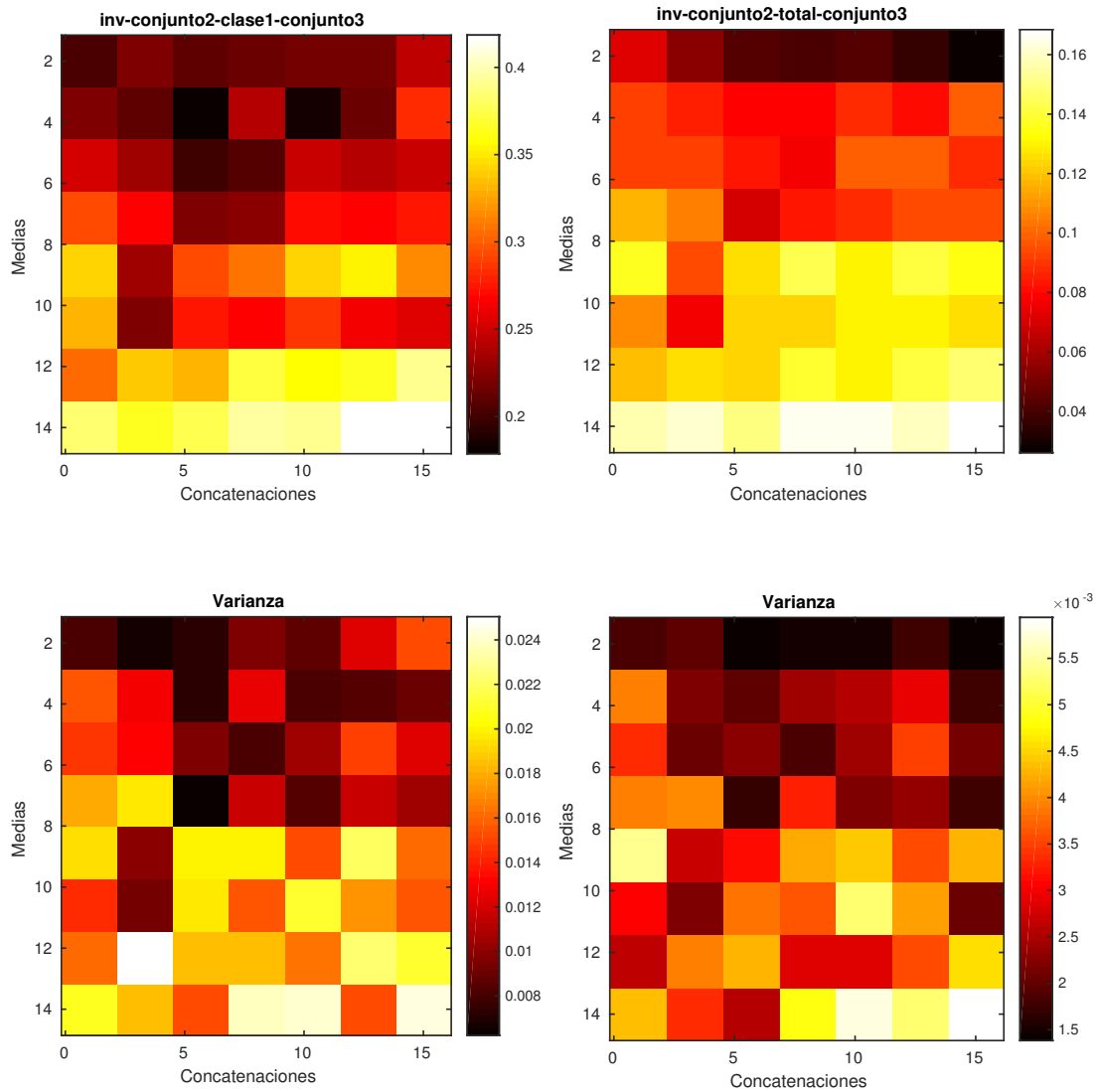


Figura A.17: Mapa de color del subconjunto 2 del método zlib. El coeficiente representado a la izquierda representa la clase del ERP P300, y el de la derecha el total.

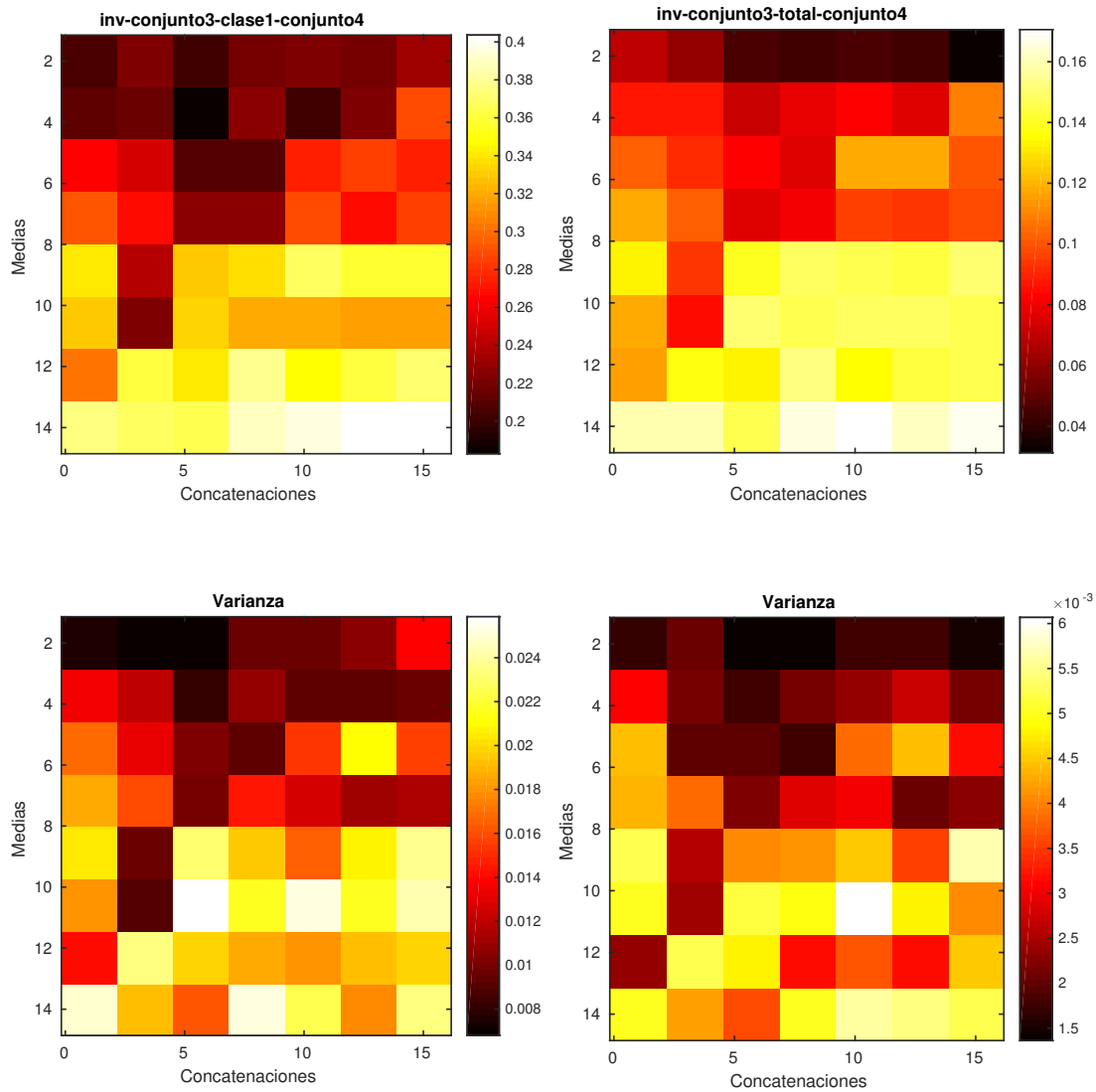


Figura A.18: Mapa de color del subconjunto 3 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

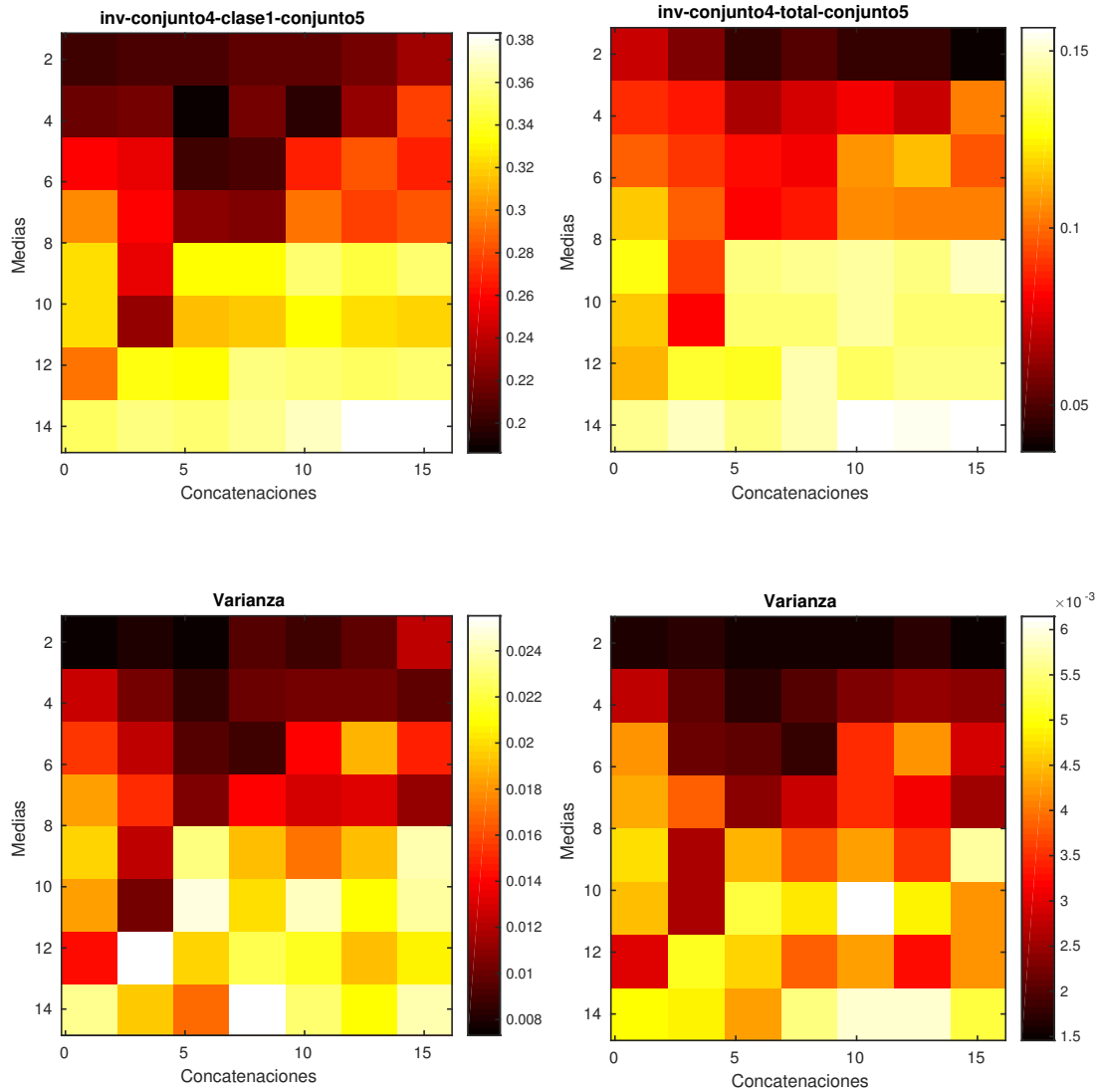


Figura A.19: Mapa de color del subconjunto 4 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

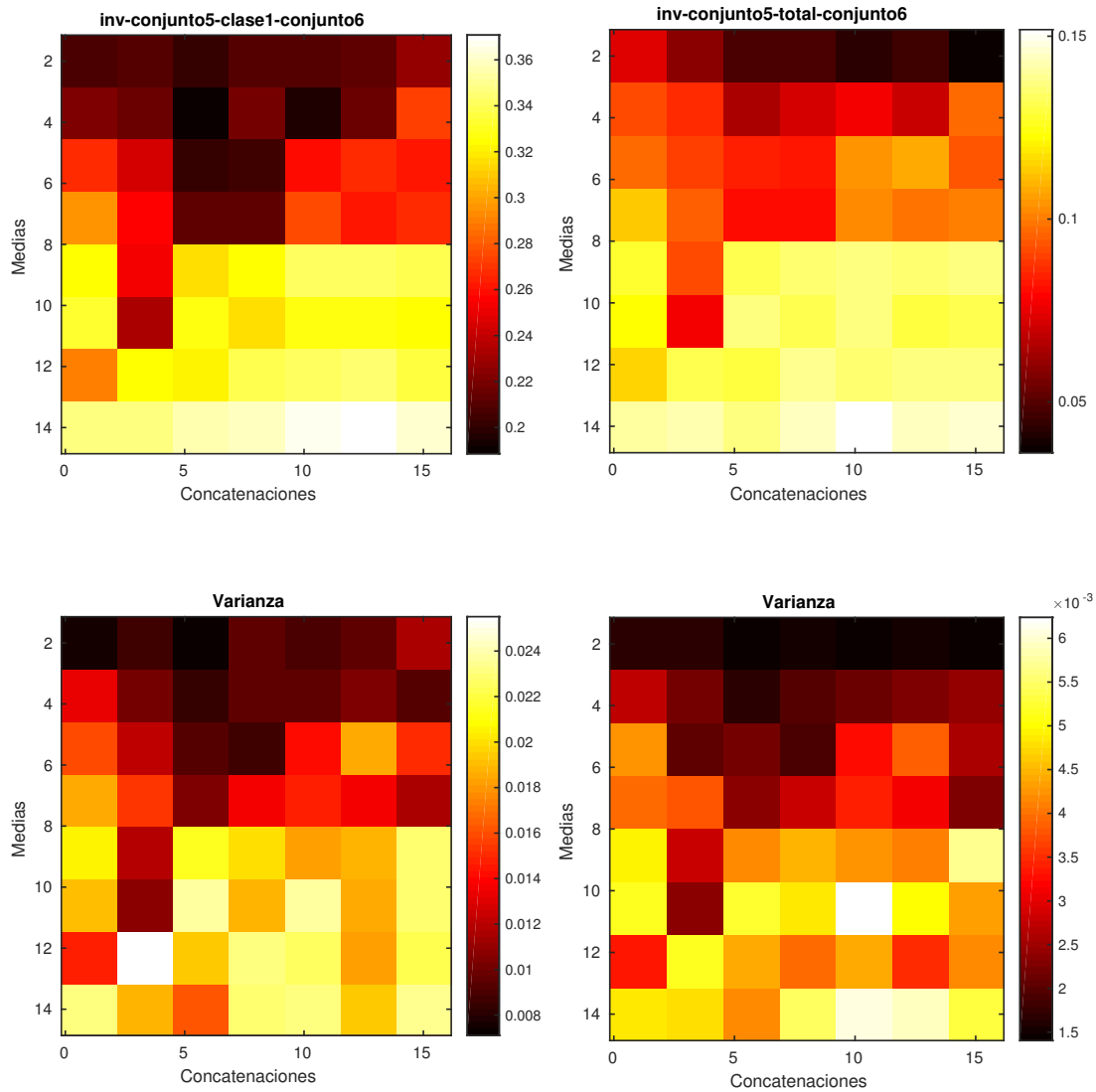


Figura A.20: Mapa de color del subconjunto 5 del método zlib. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.5. Mx: conjunto A, subconjuntos del 1 al 5

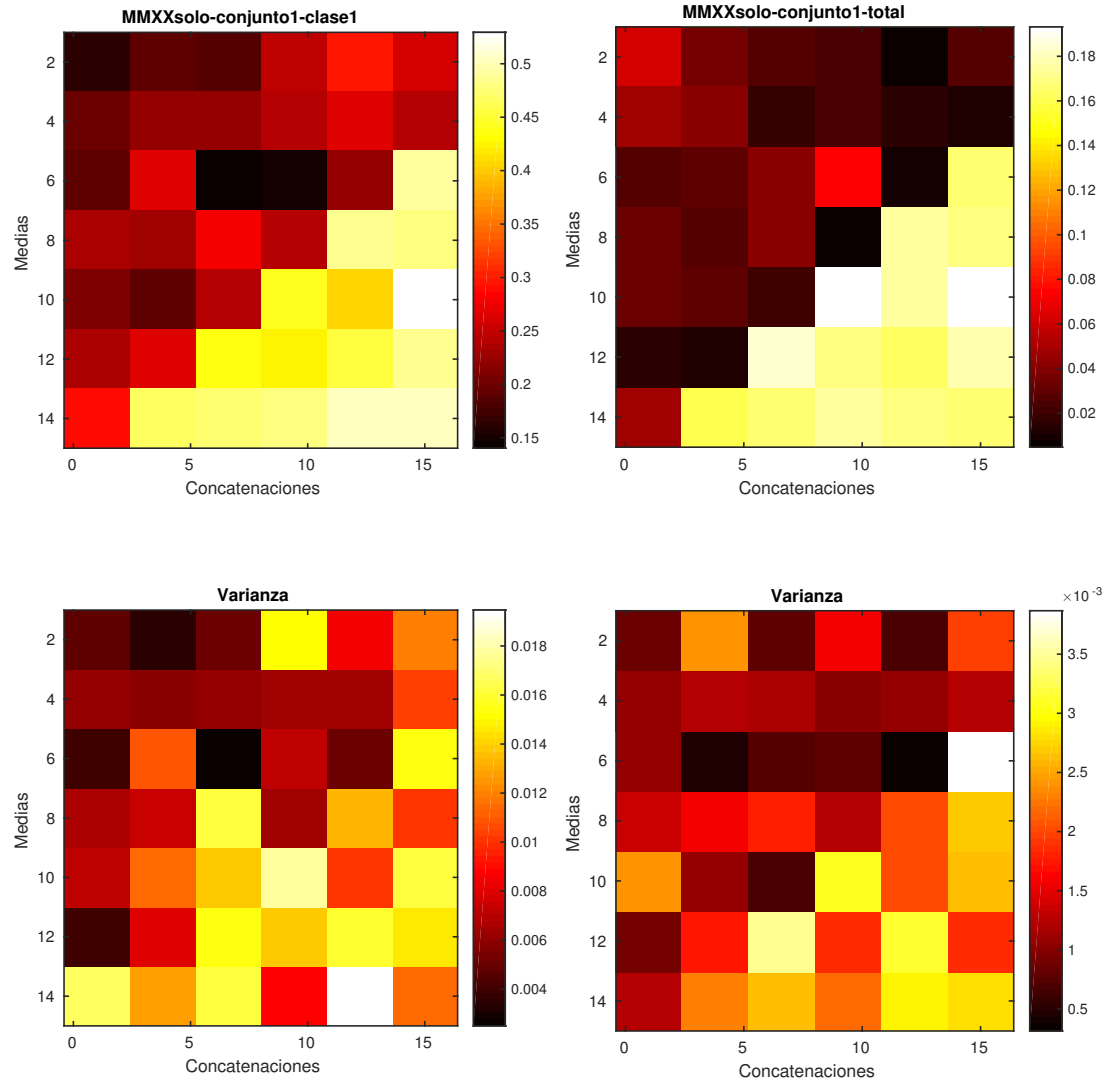


Figura A.21: Mapa de color del subconjunto 1 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

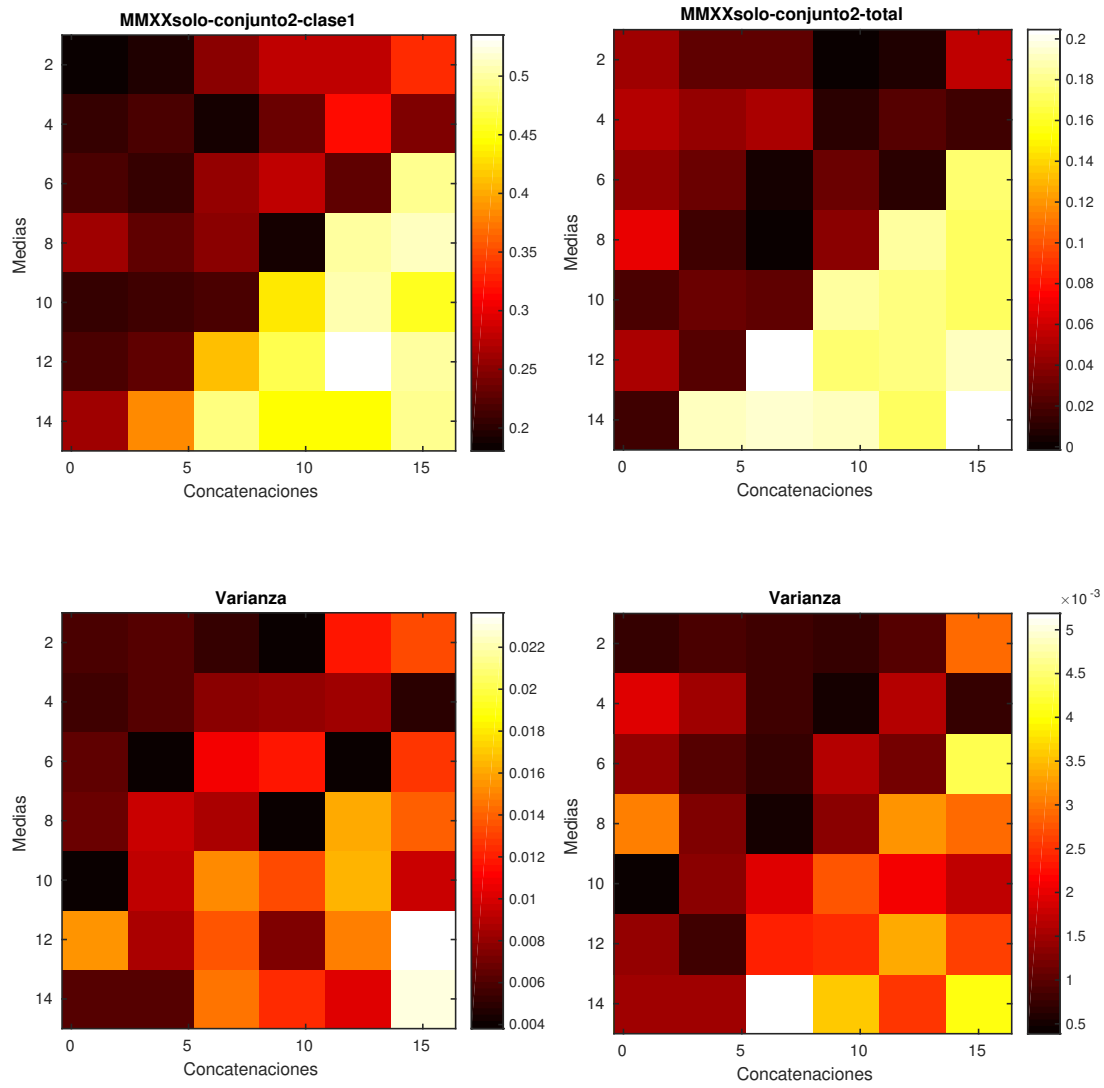


Figura A.22: Mapa de color del subconjunto 2 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

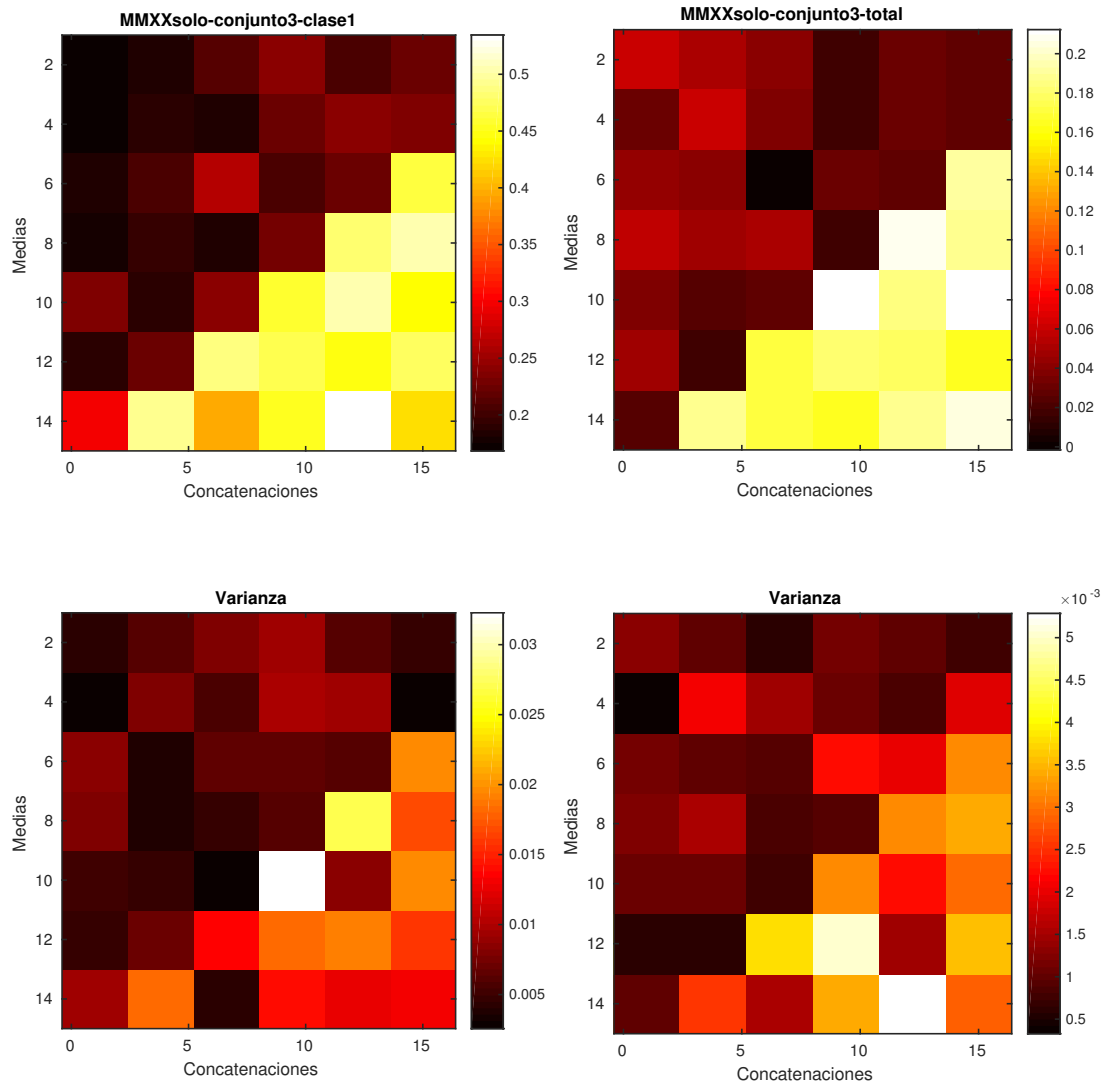


Figura A.23: Mapa de color del subconjunto 3 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

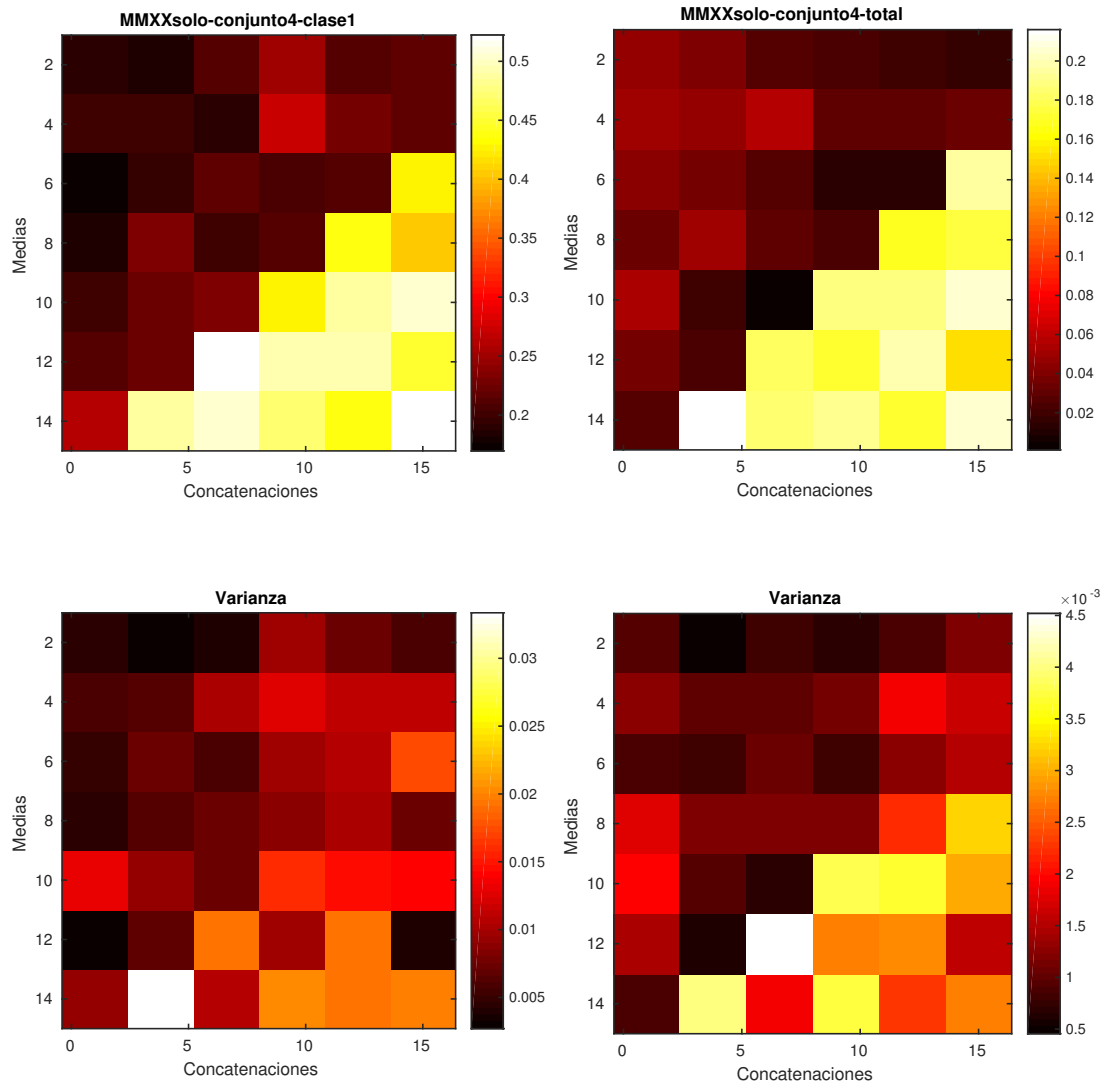


Figura A.24: Mapa de color del subconjunto 4 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

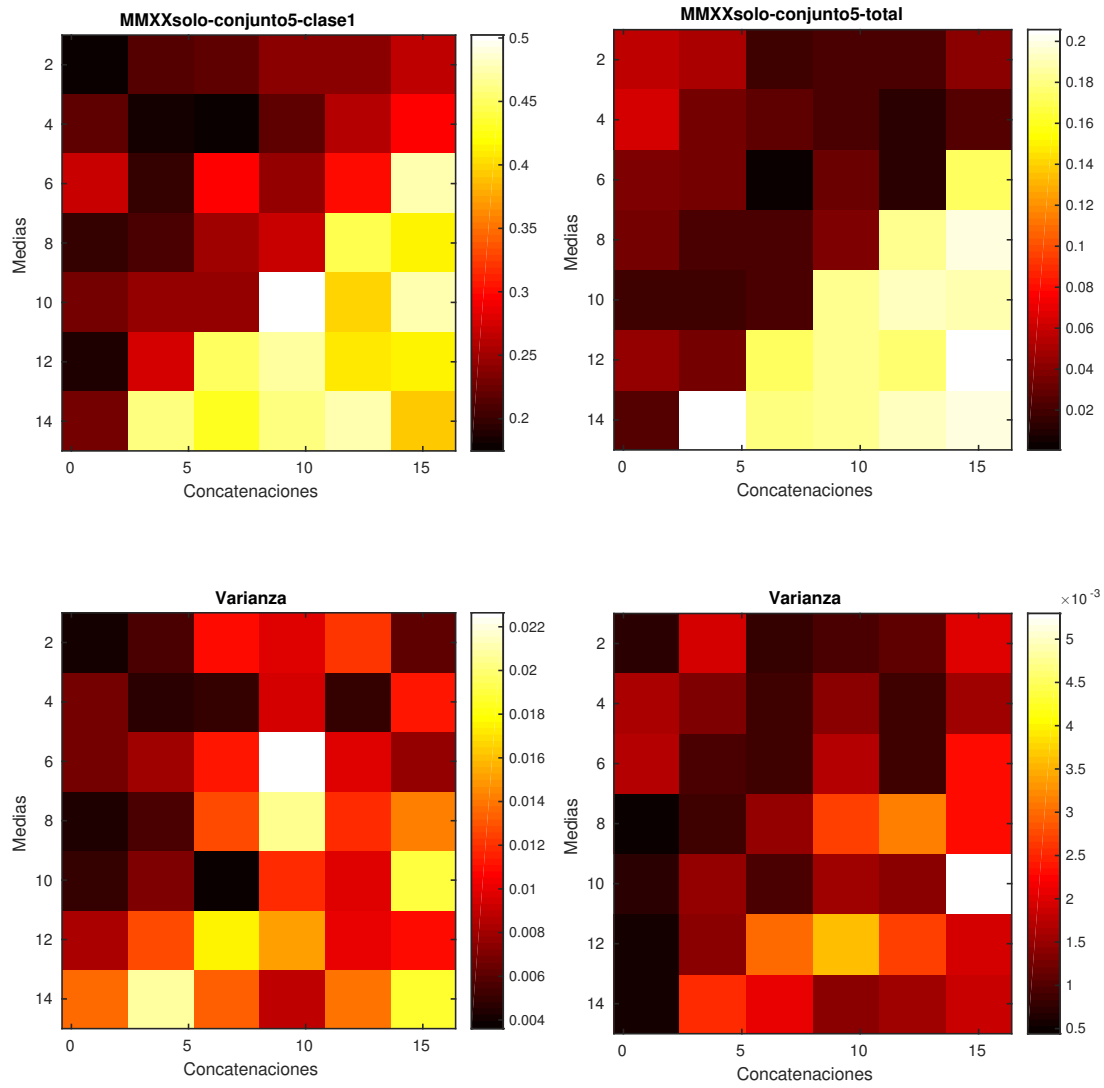


Figura A.25: Mapa de color del subconjunto 5 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.6. Mx: conjunto A, subconjuntos del 1 al 5 por invasión

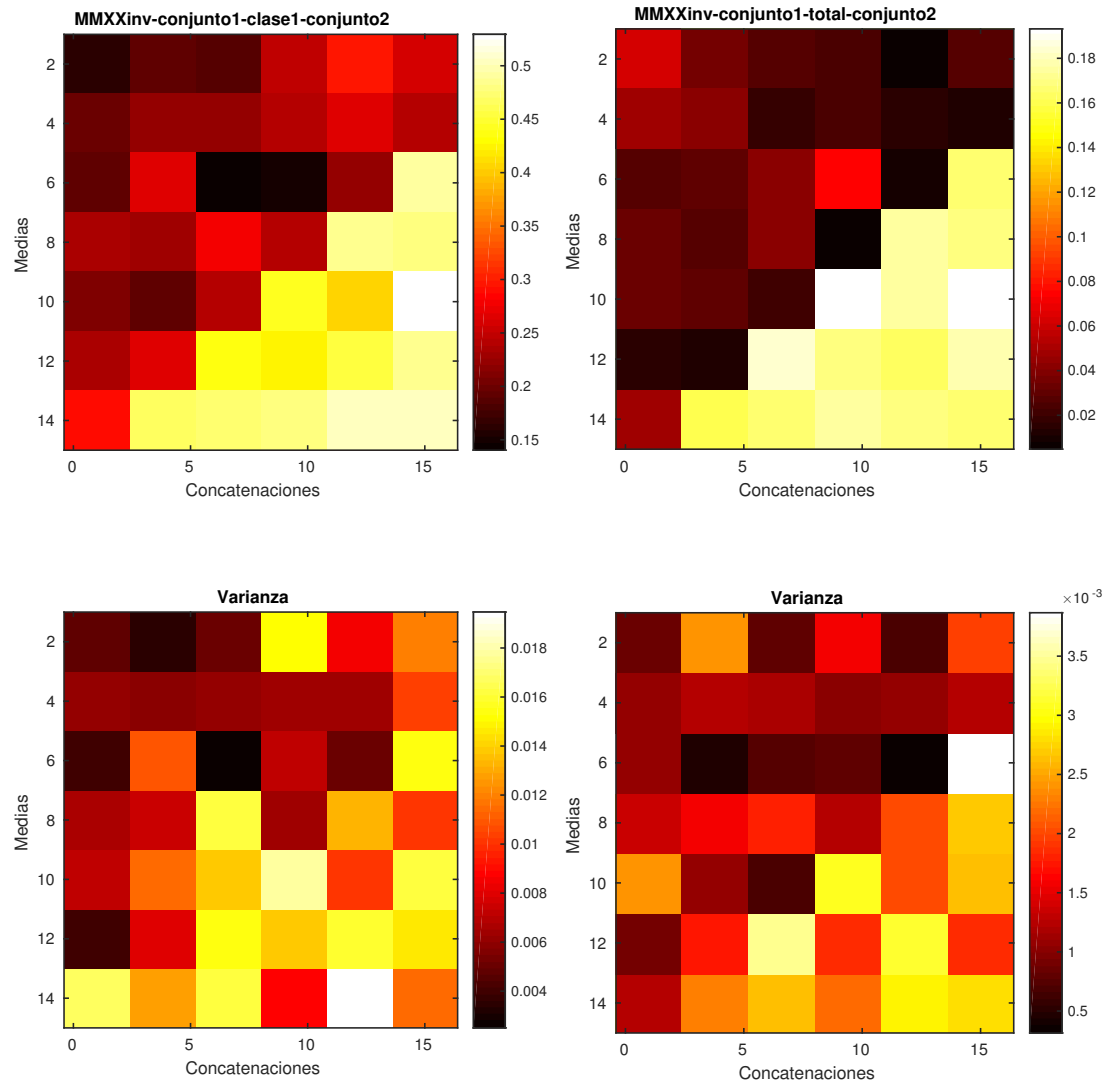


Figura A.26: Mapa de color del subconjunto 1 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

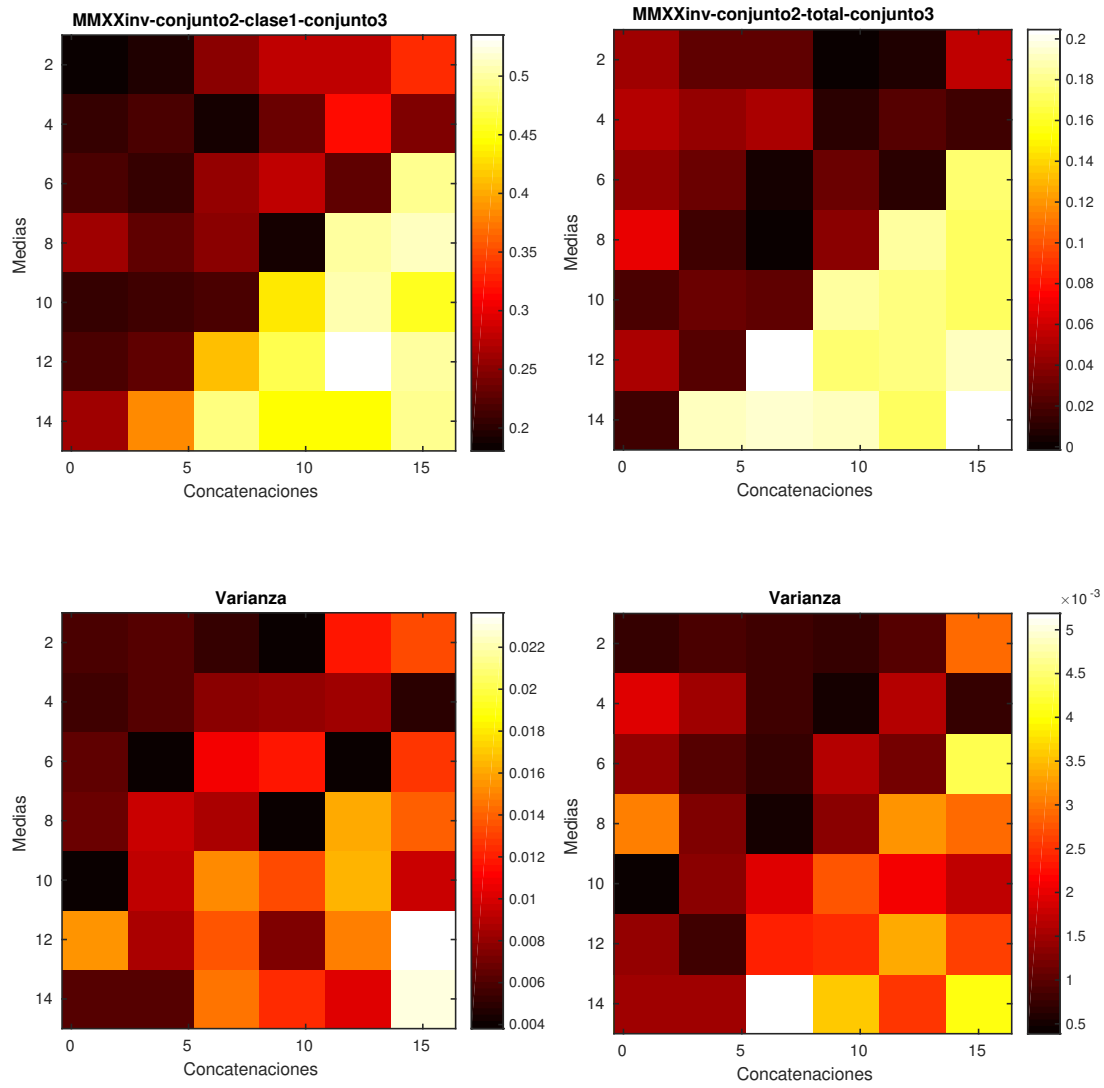


Figura A.27: Mapa de color del subconjunto 2 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

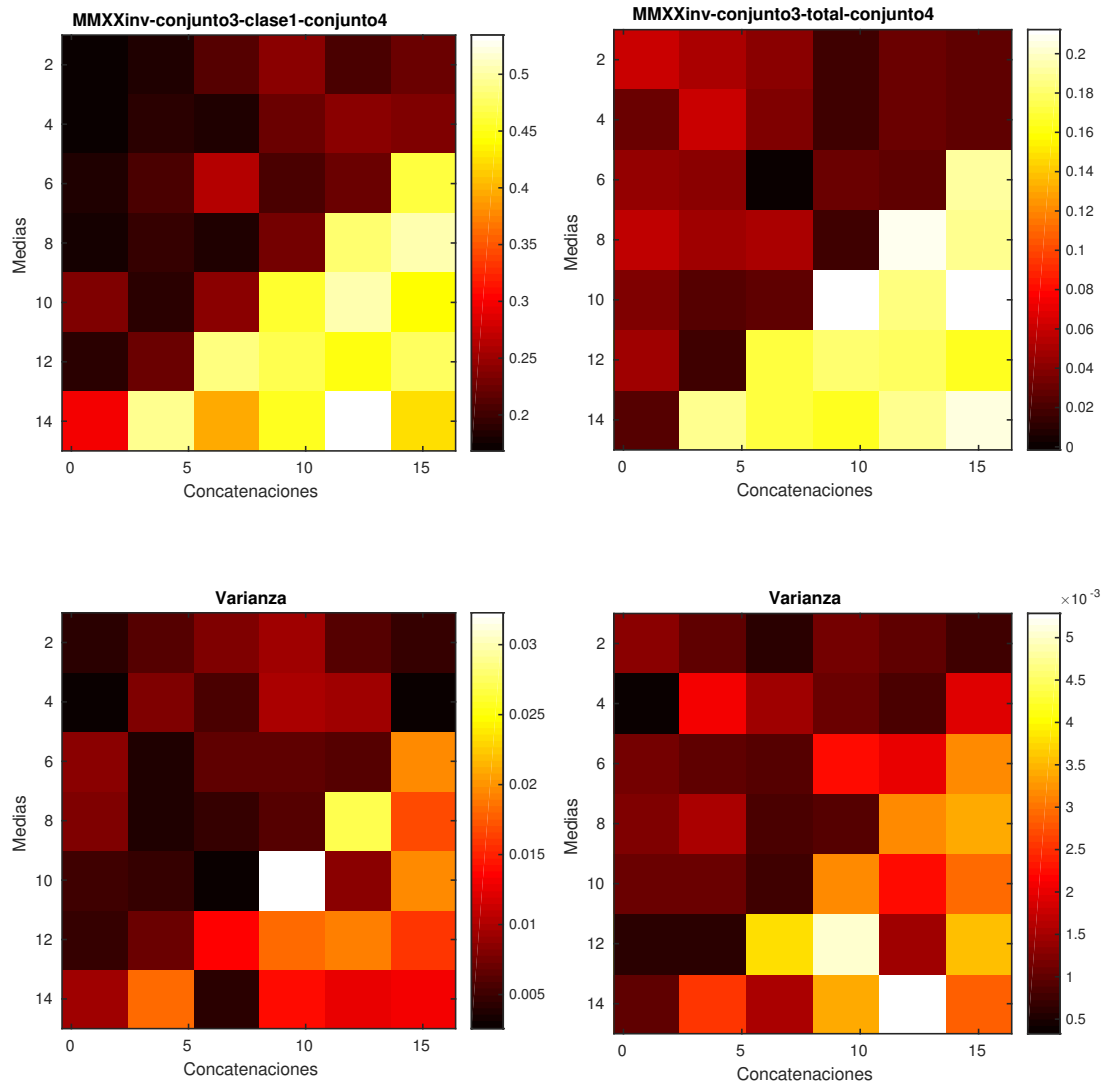


Figura A.28: Mapa de color del subconjunto 3 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

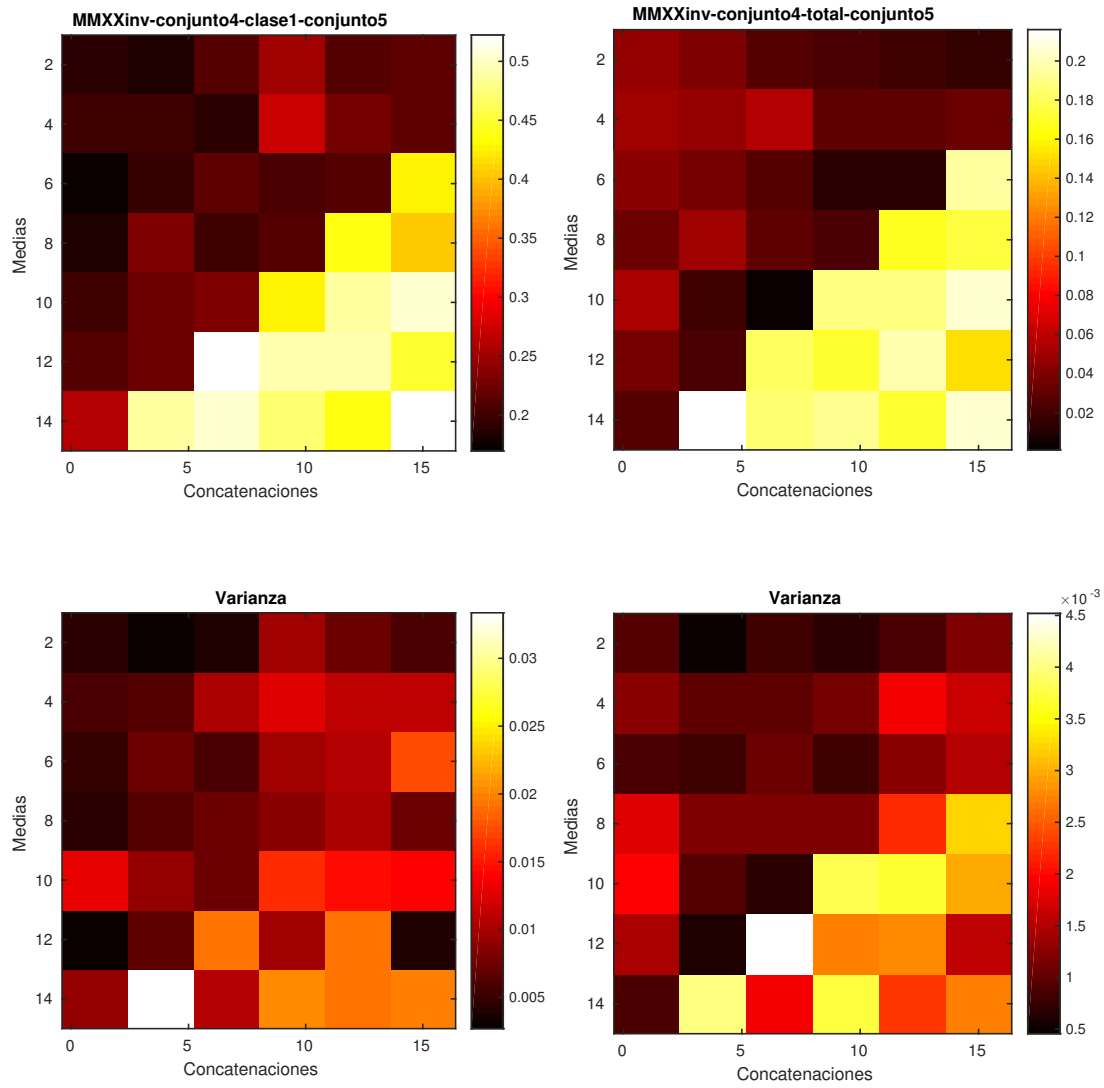


Figura A.29: Mapa de color del subconjunto 4 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

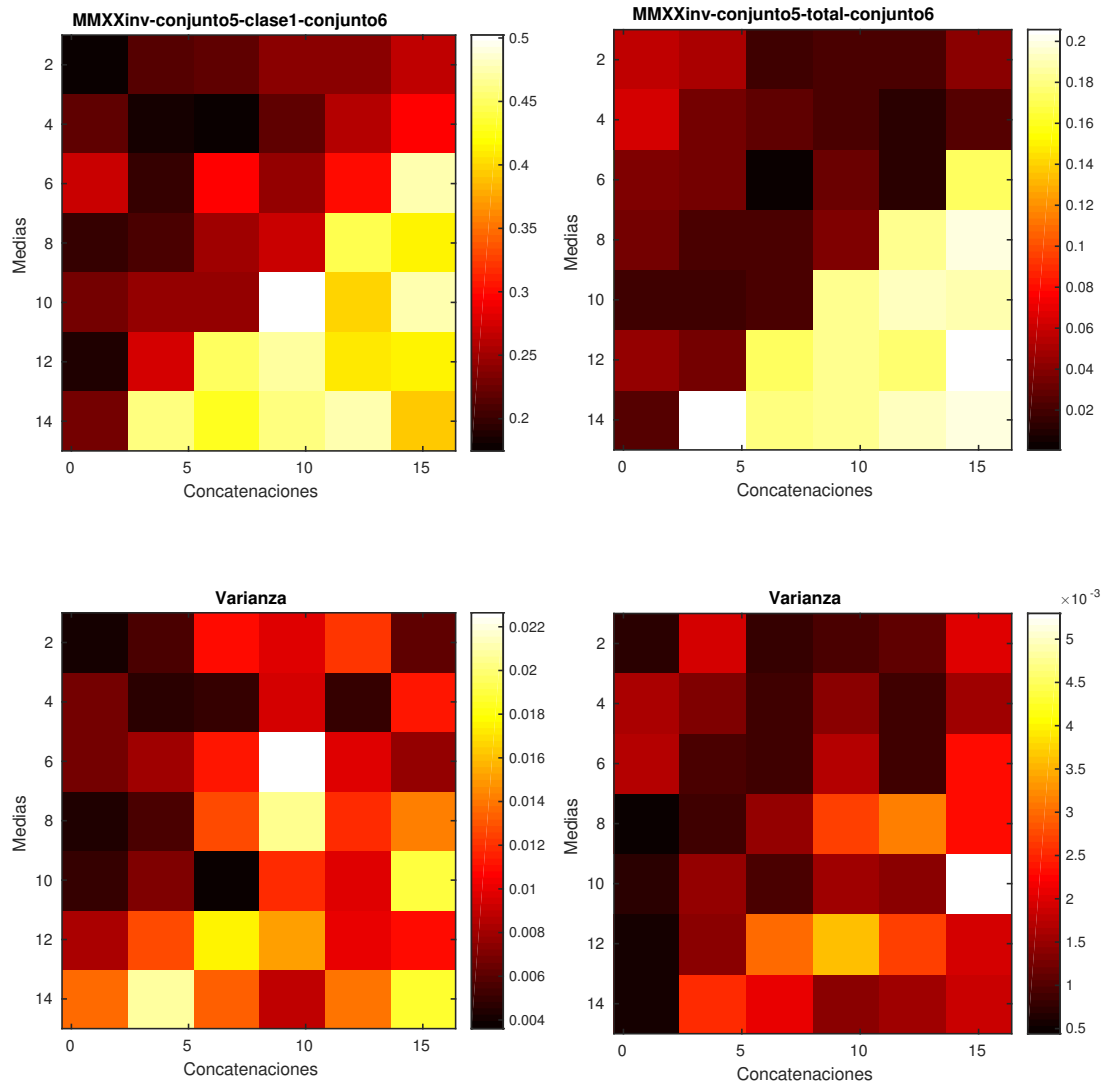


Figura A.30: Mapa de color del subconjunto 5 del método mx. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.7. Std: conjunto A, subconjuntos del 1 al 5

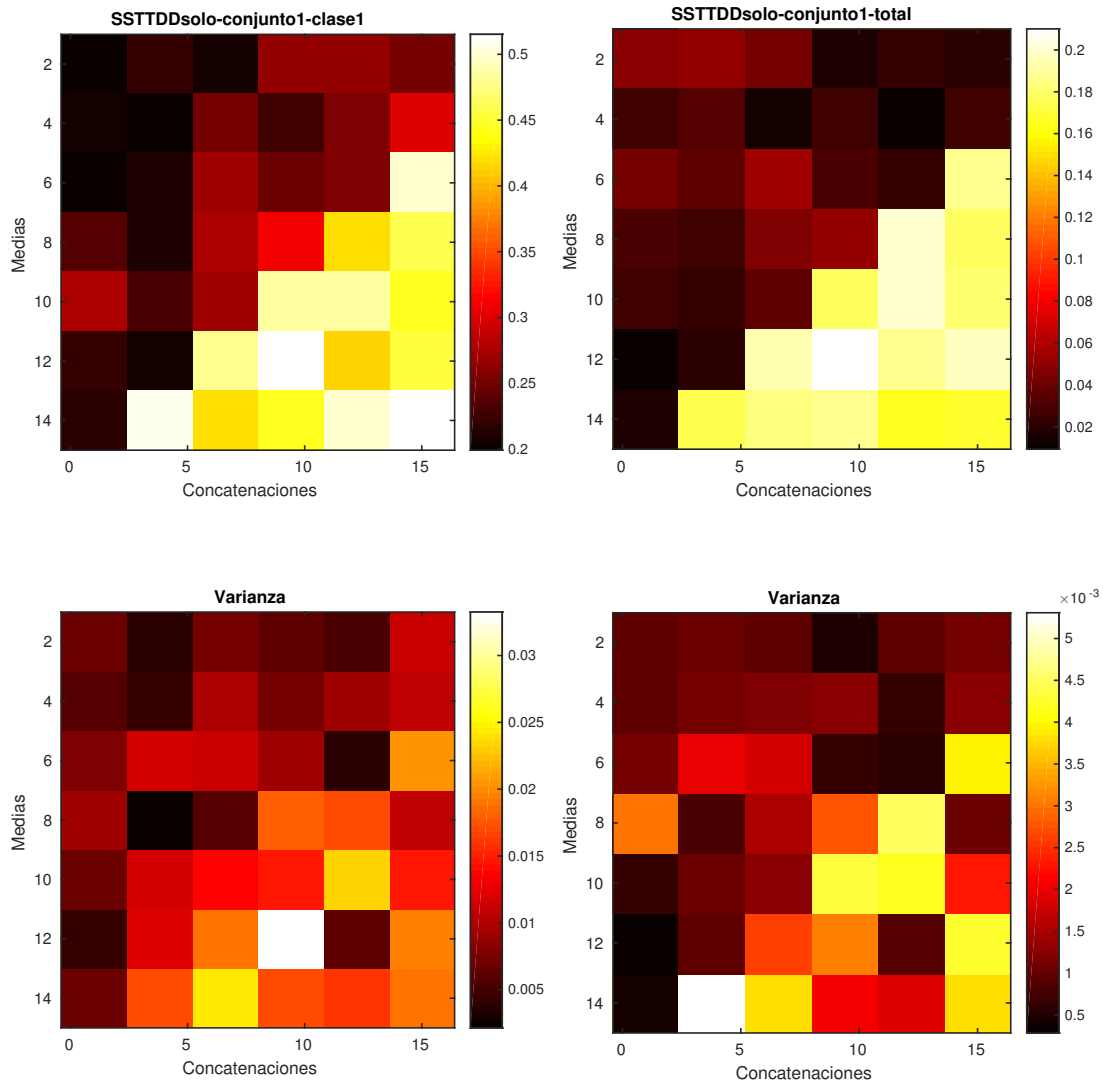


Figura A.31: Mapa de color del subconjunto 1 del método std. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.

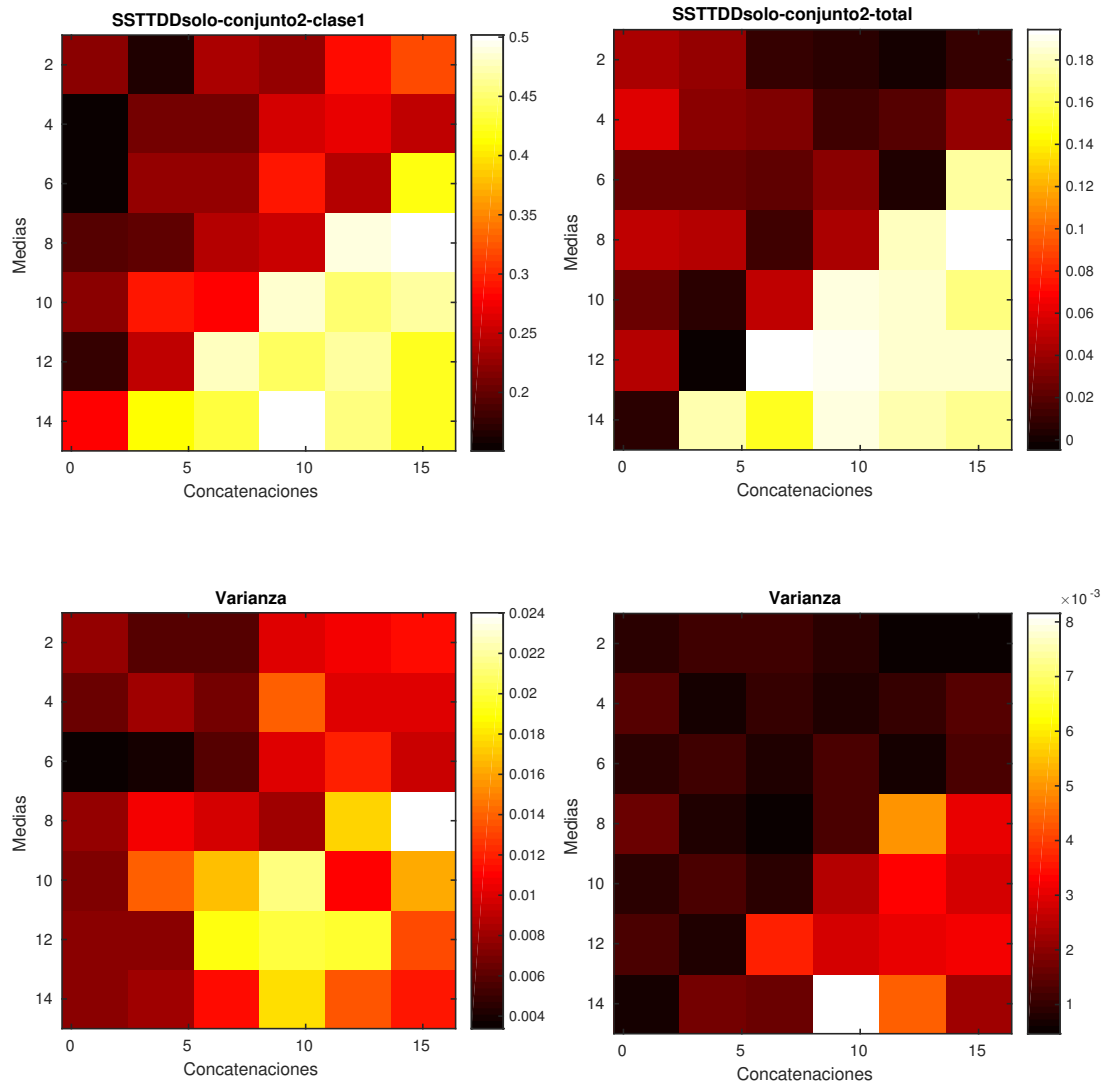


Figura A.32: Mapa de color del subconjunto 2 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

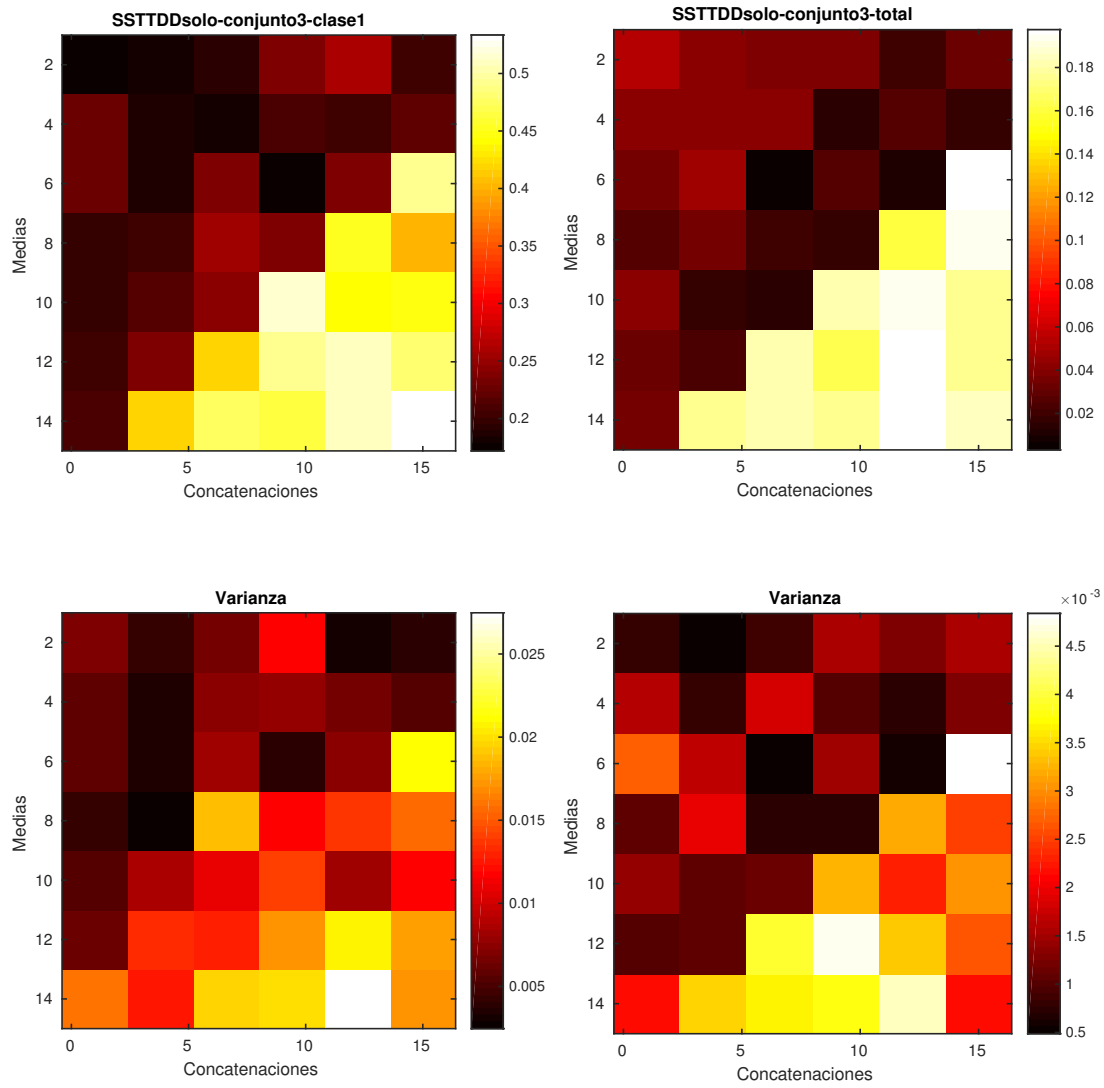


Figura A.33: Mapa de color del subconjunto 3 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

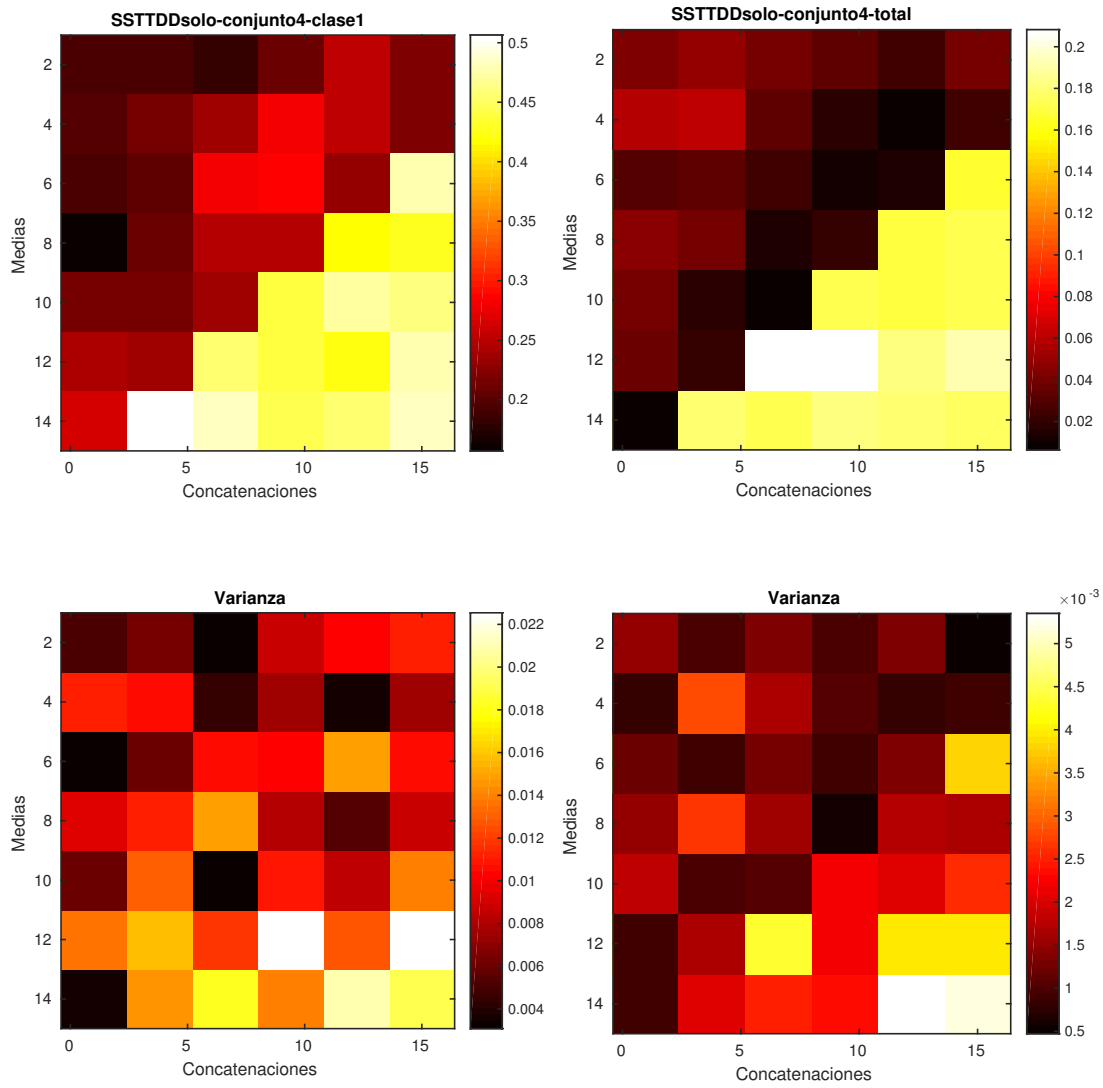


Figura A.34: Mapa de color del subconjunto 4 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

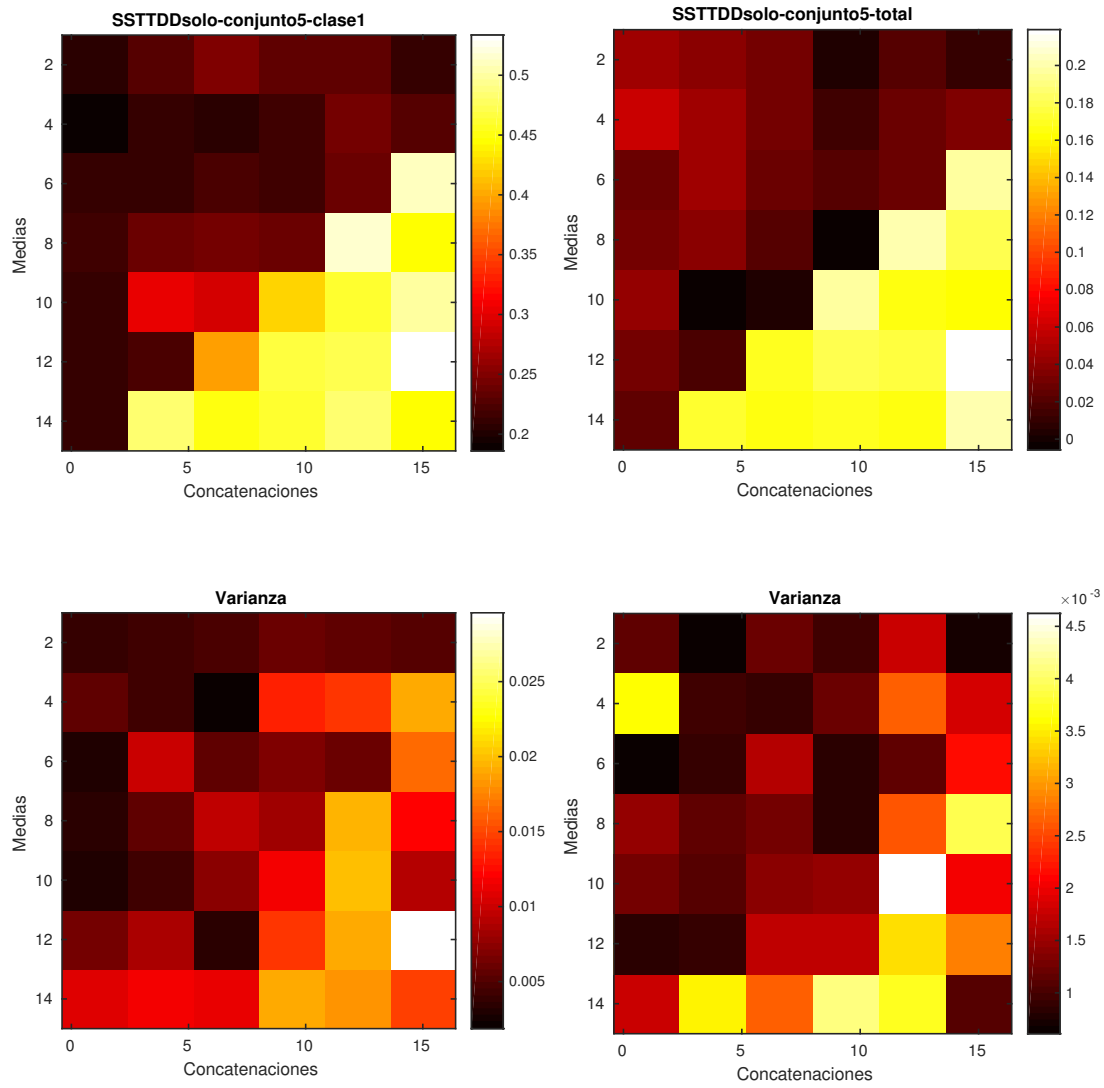


Figura A.35: Mapa de color del subconjunto 5 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.8. Std: conjunto A, subconjuntos del 1 al 5 por invasión

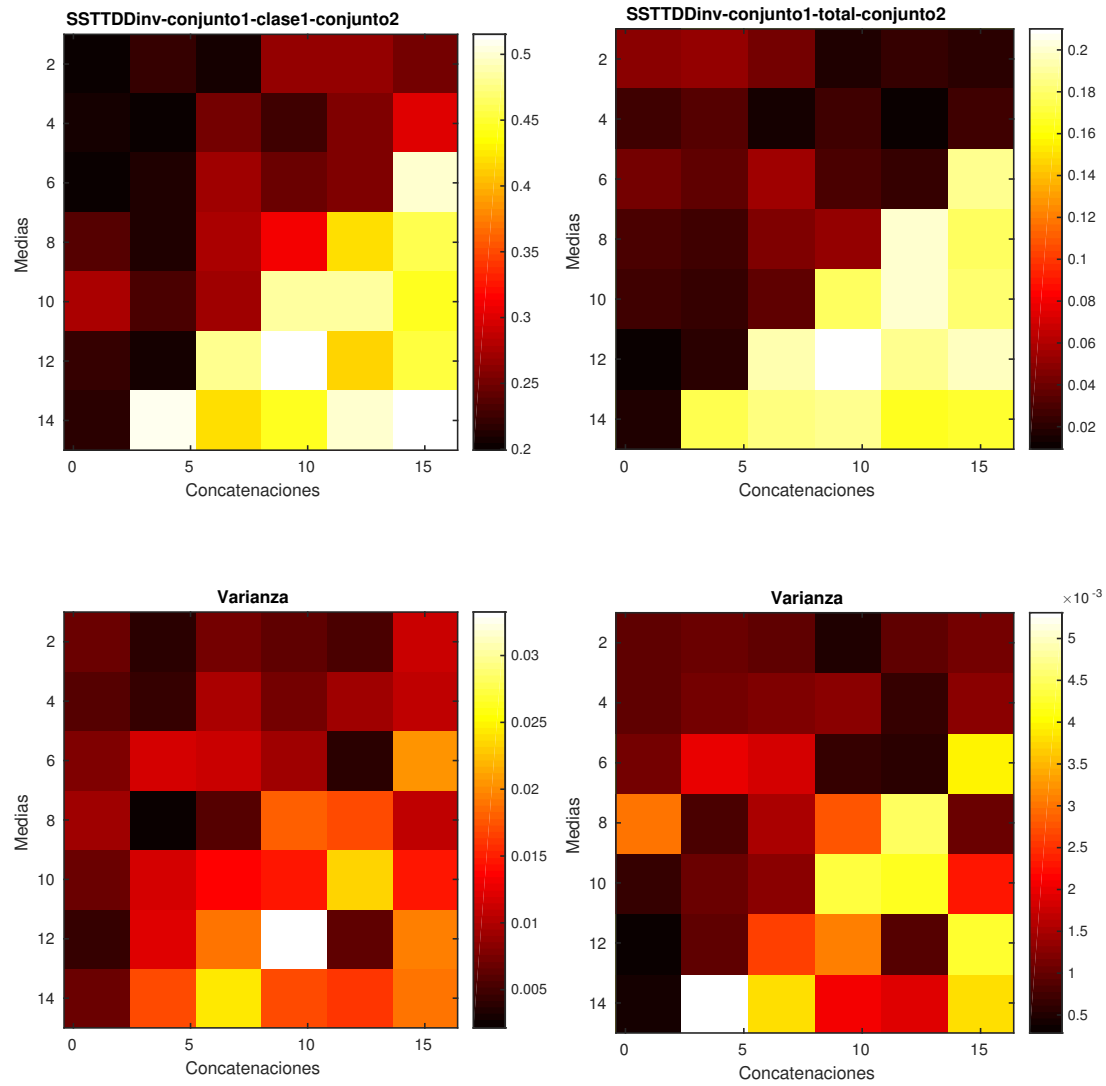


Figura A.36: Mapa de color del subconjunto 1 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

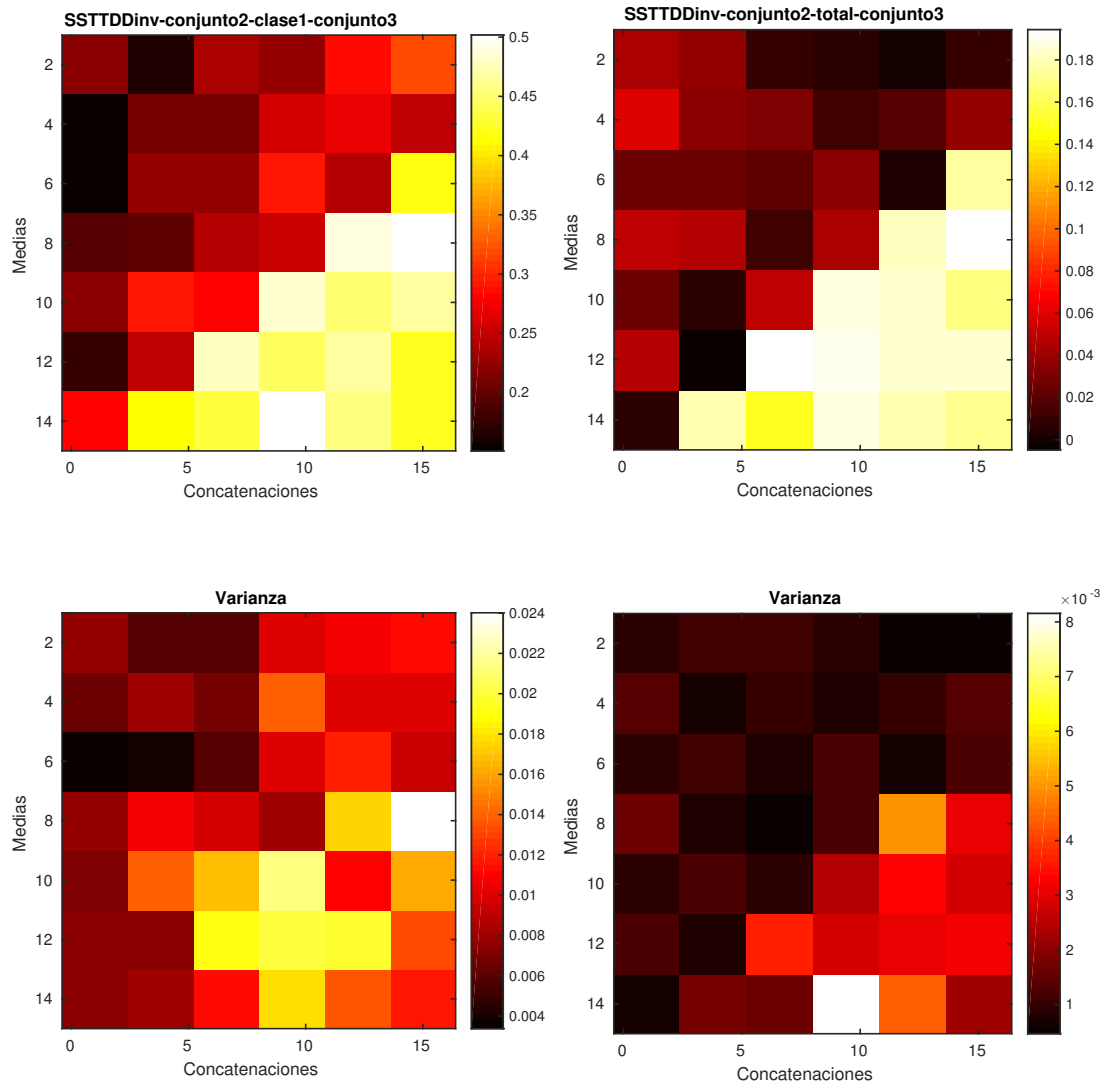


Figura A.37: Mapa de color del subconjunto 2 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

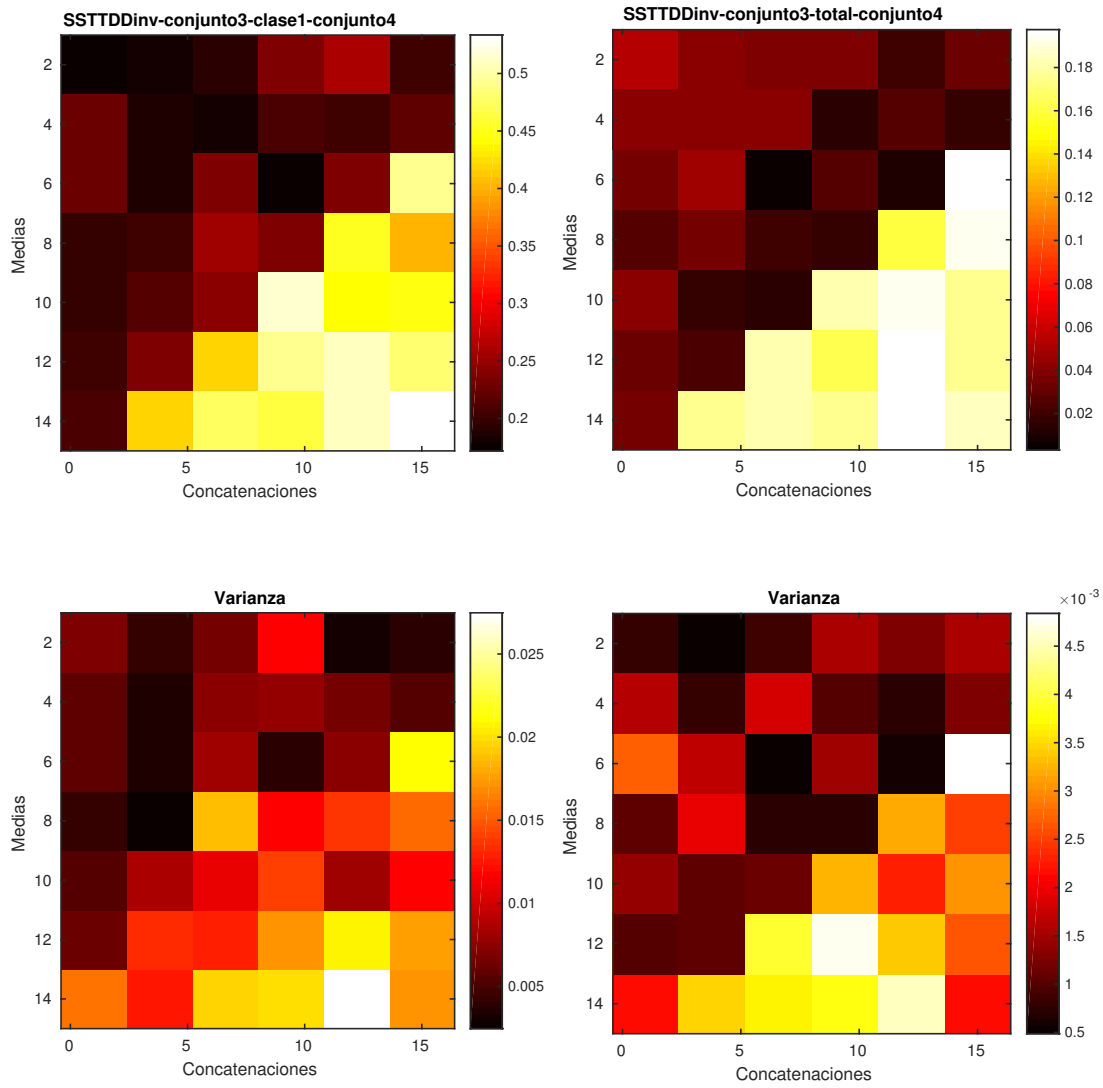


Figura A.38: Mapa de color del subconjunto 3 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

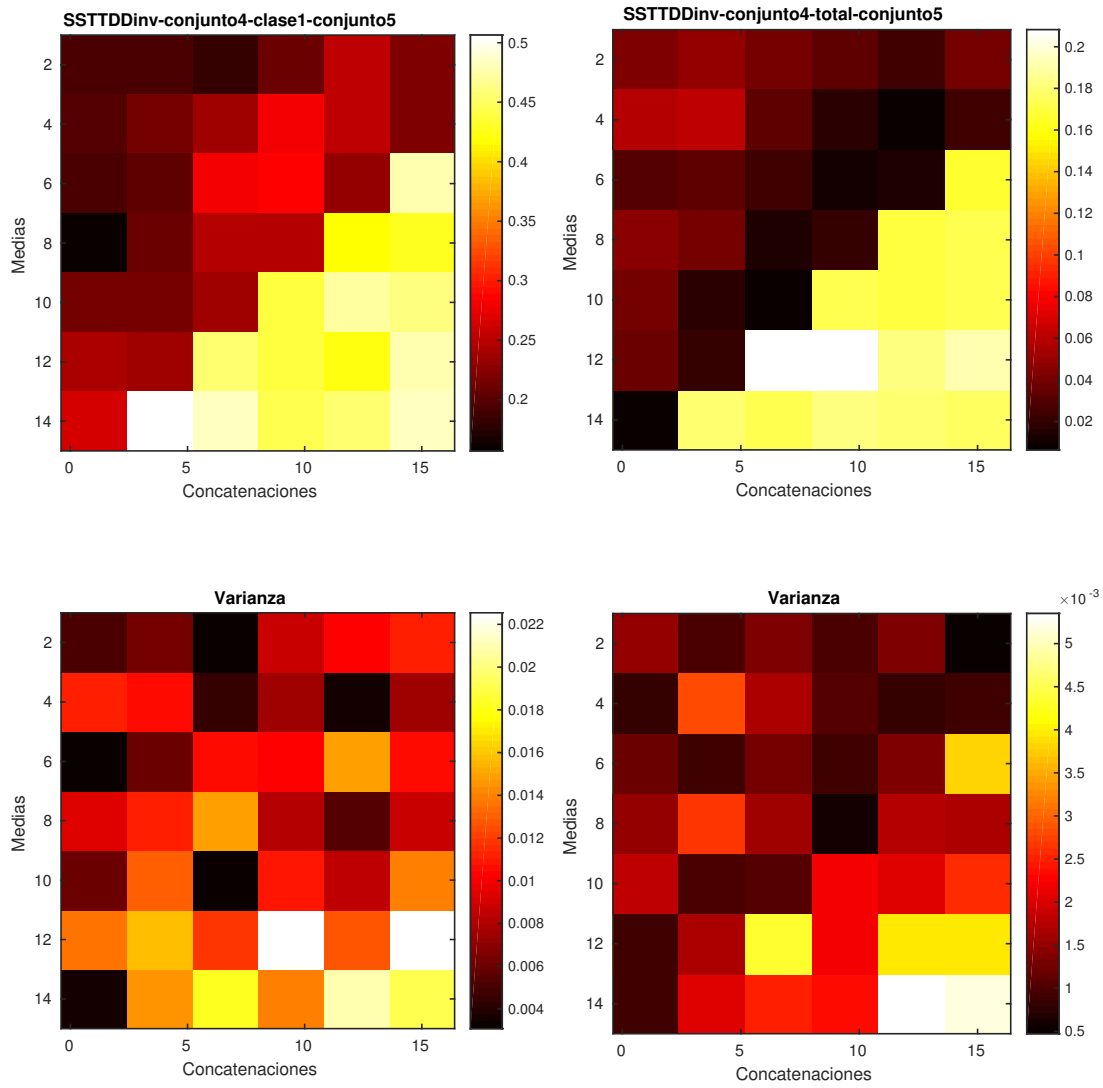


Figura A.39: Mapa de color del subconjunto 4 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

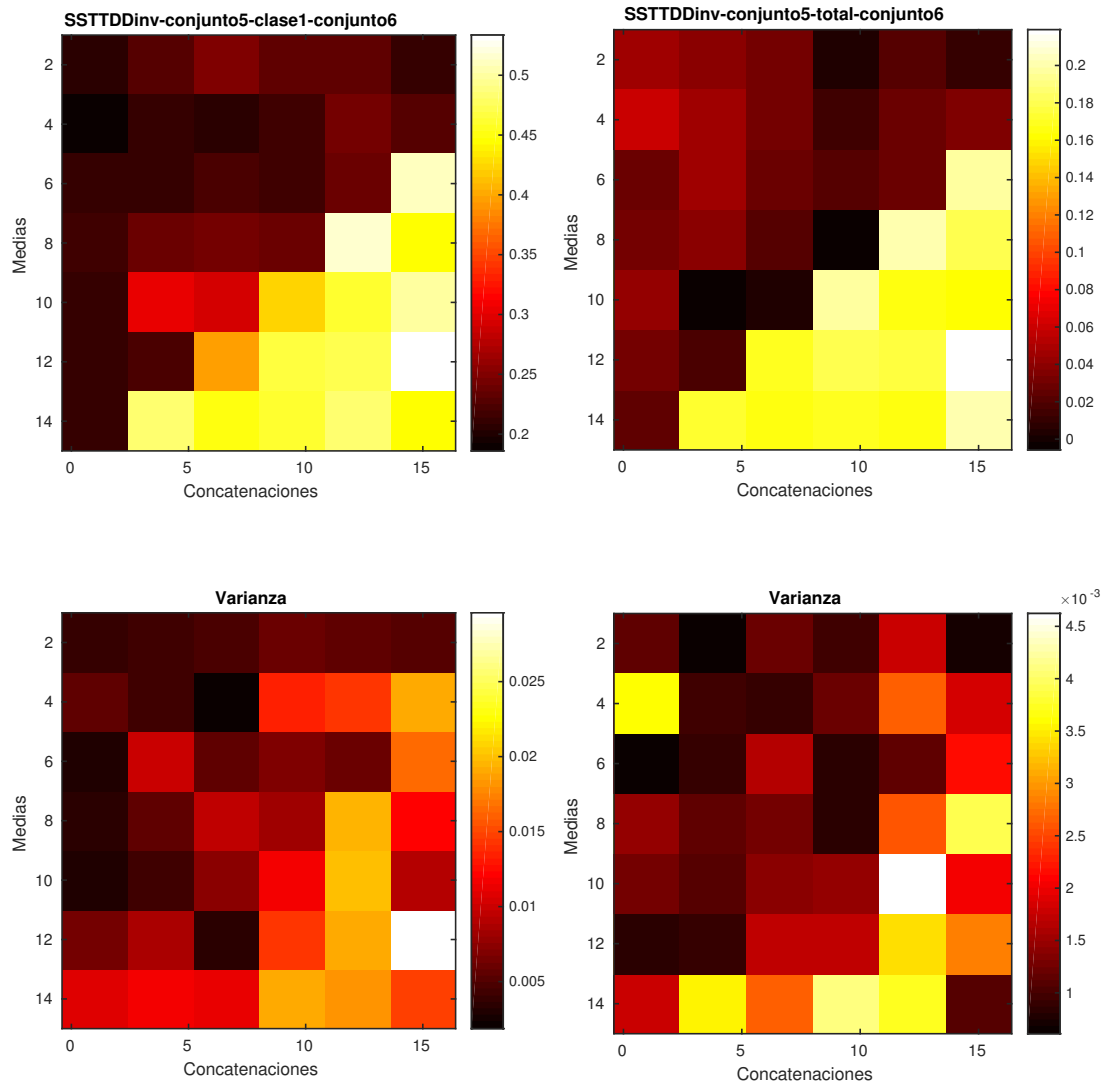


Figura A.40: Mapa de color del subconjunto 5 del método std. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.9. Sl: conjunto A, subconjuntos del 1 al 5

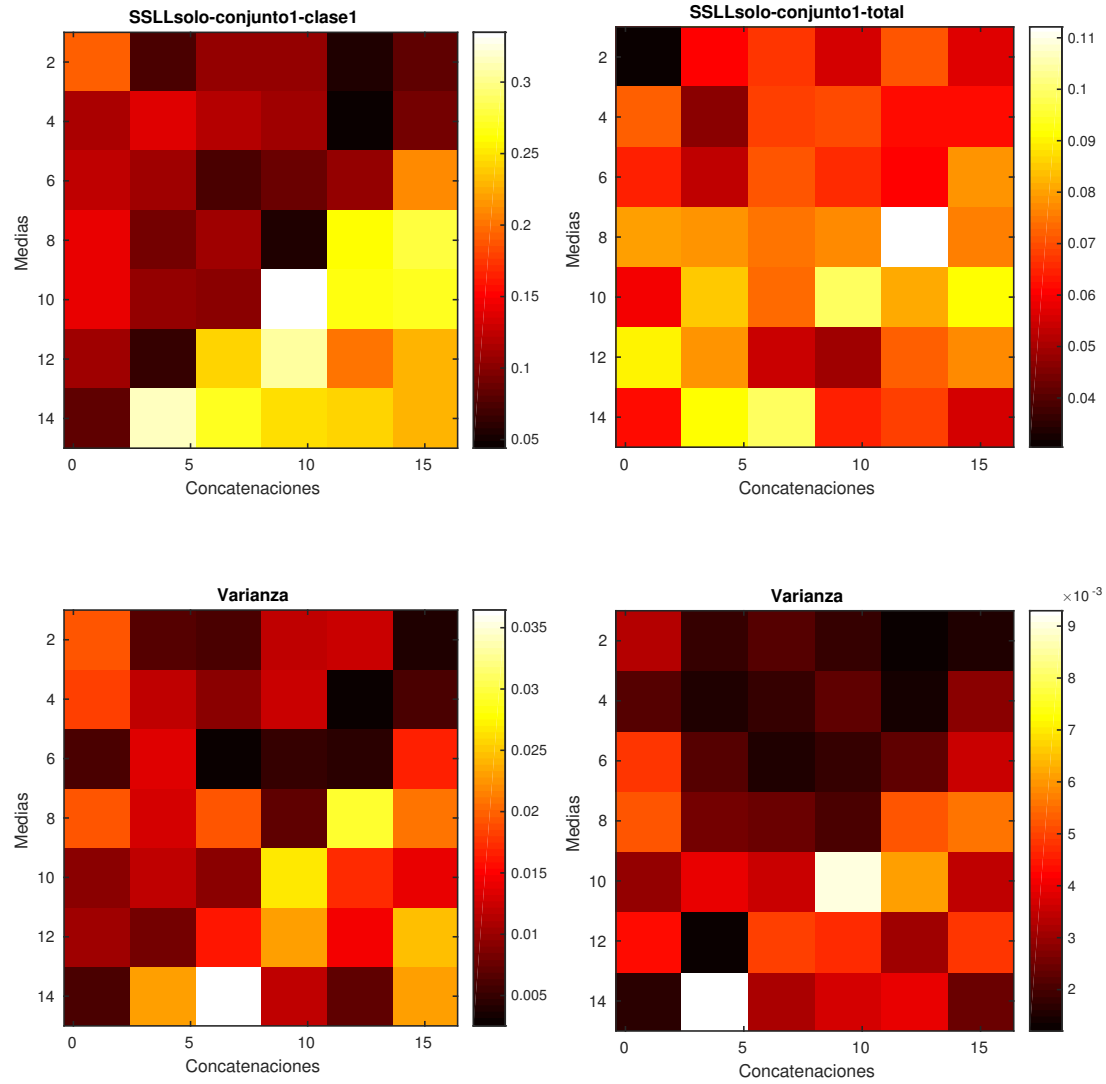


Figura A.41: Mapa de color del subconjunto 1 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

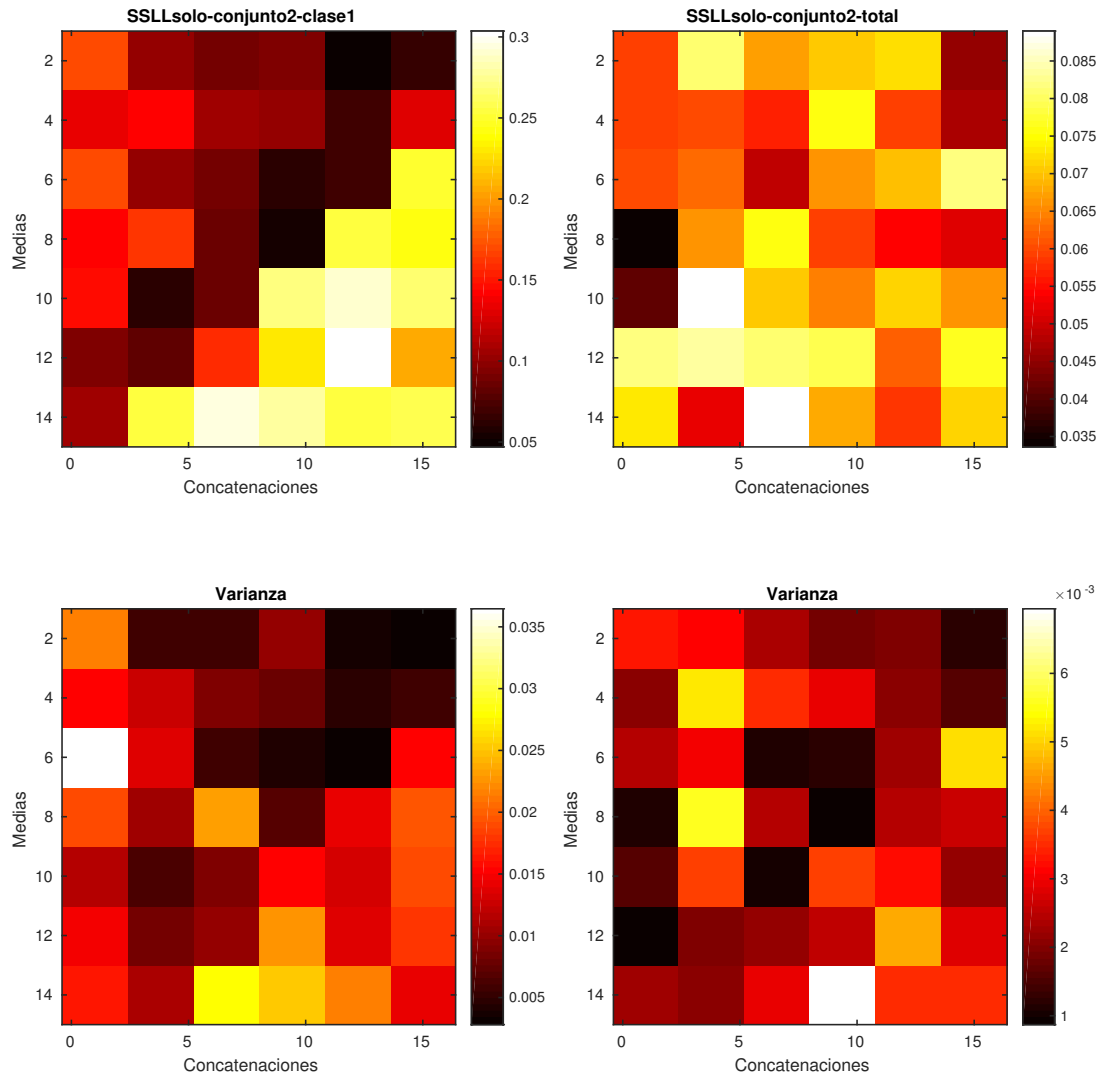


Figura A.42: Mapa de color del subconjunto 2 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.

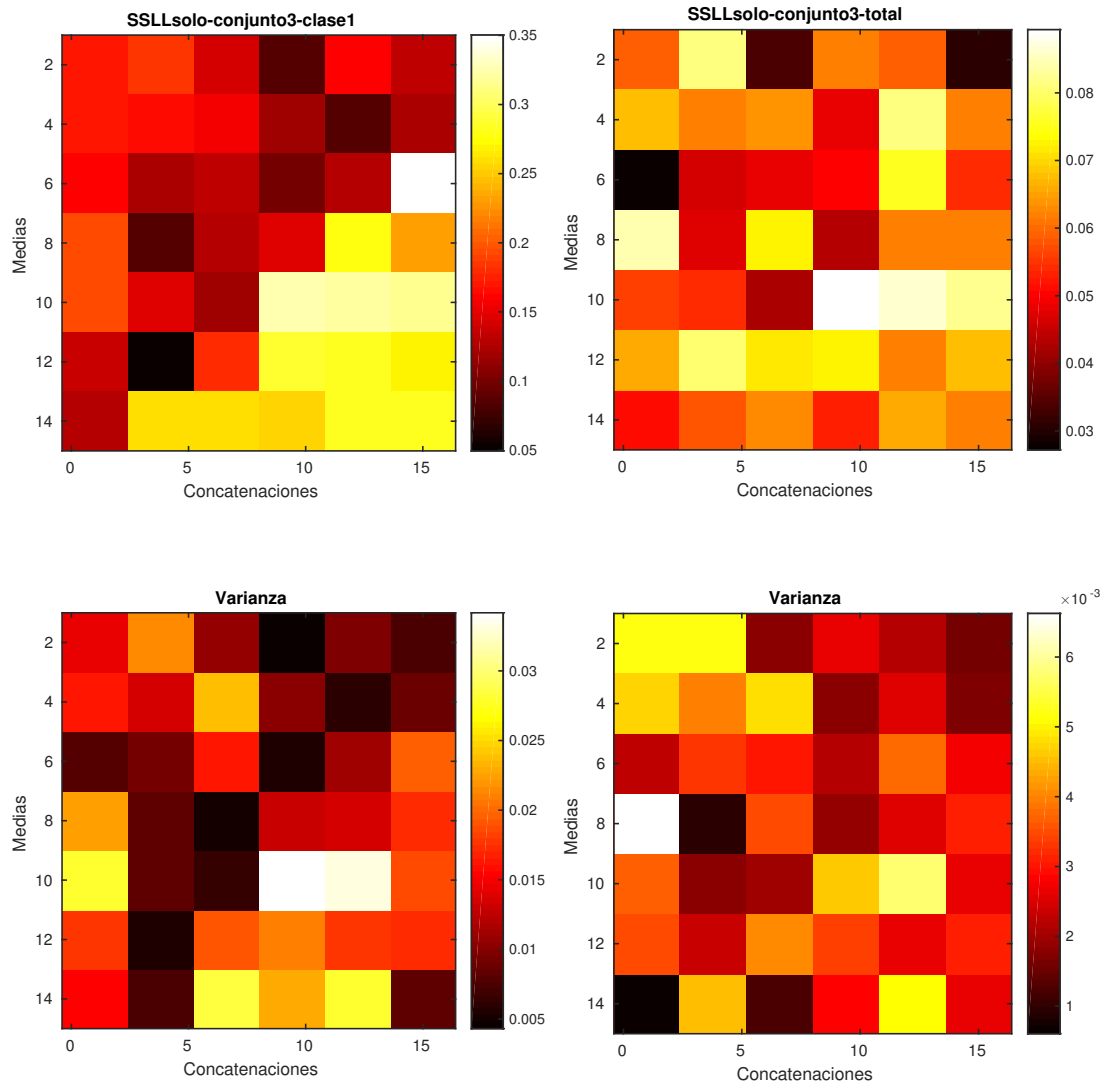


Figura A.43: Mapa de color del subconjunto 3 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

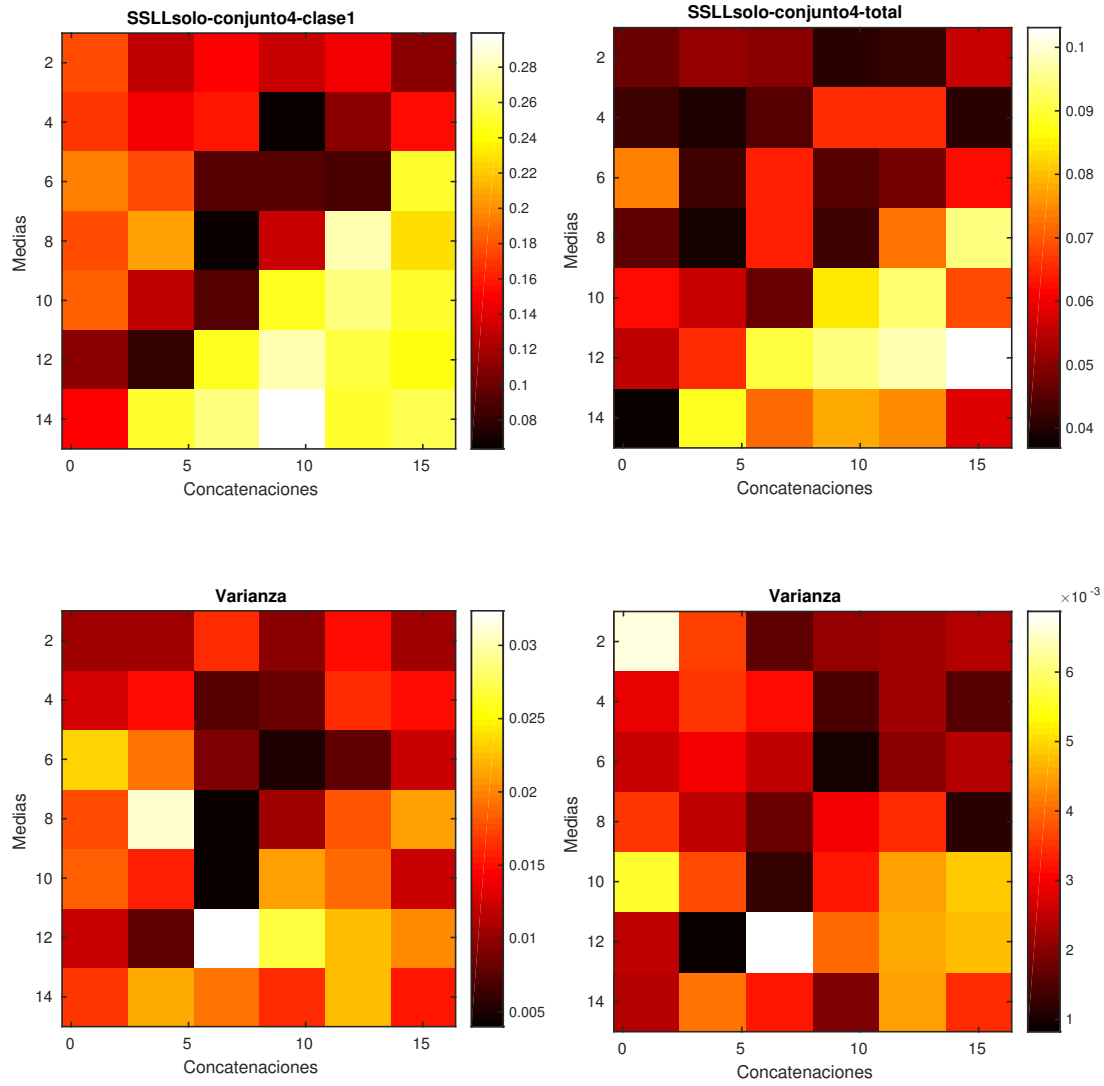


Figura A.44: Mapa de color del subconjunto 4 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

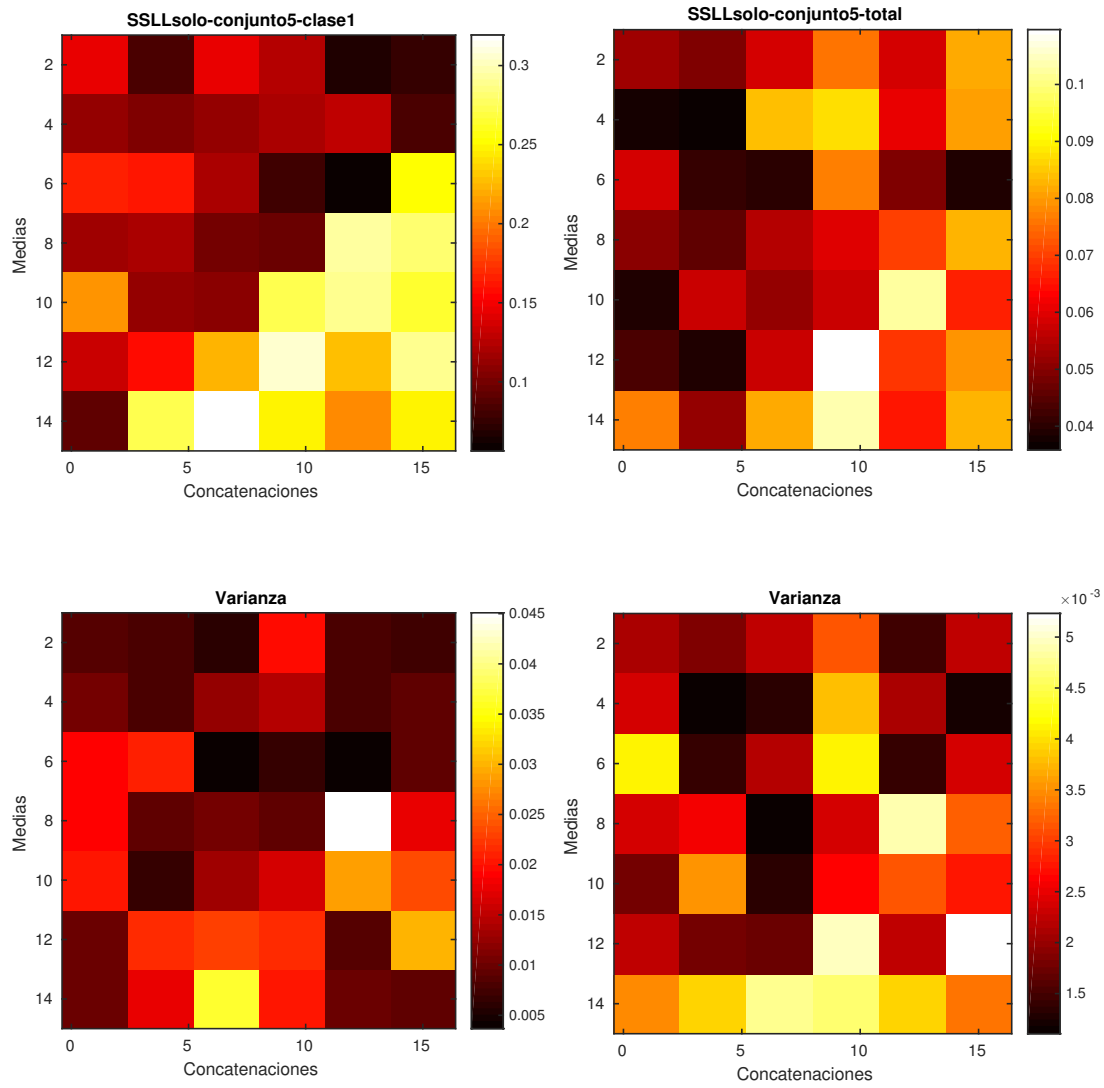


Figura A.45: Mapa de color del subconjunto 5 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

A.1.10. Sl: conjunto A, subconjuntos del 1 al 5 por invasión

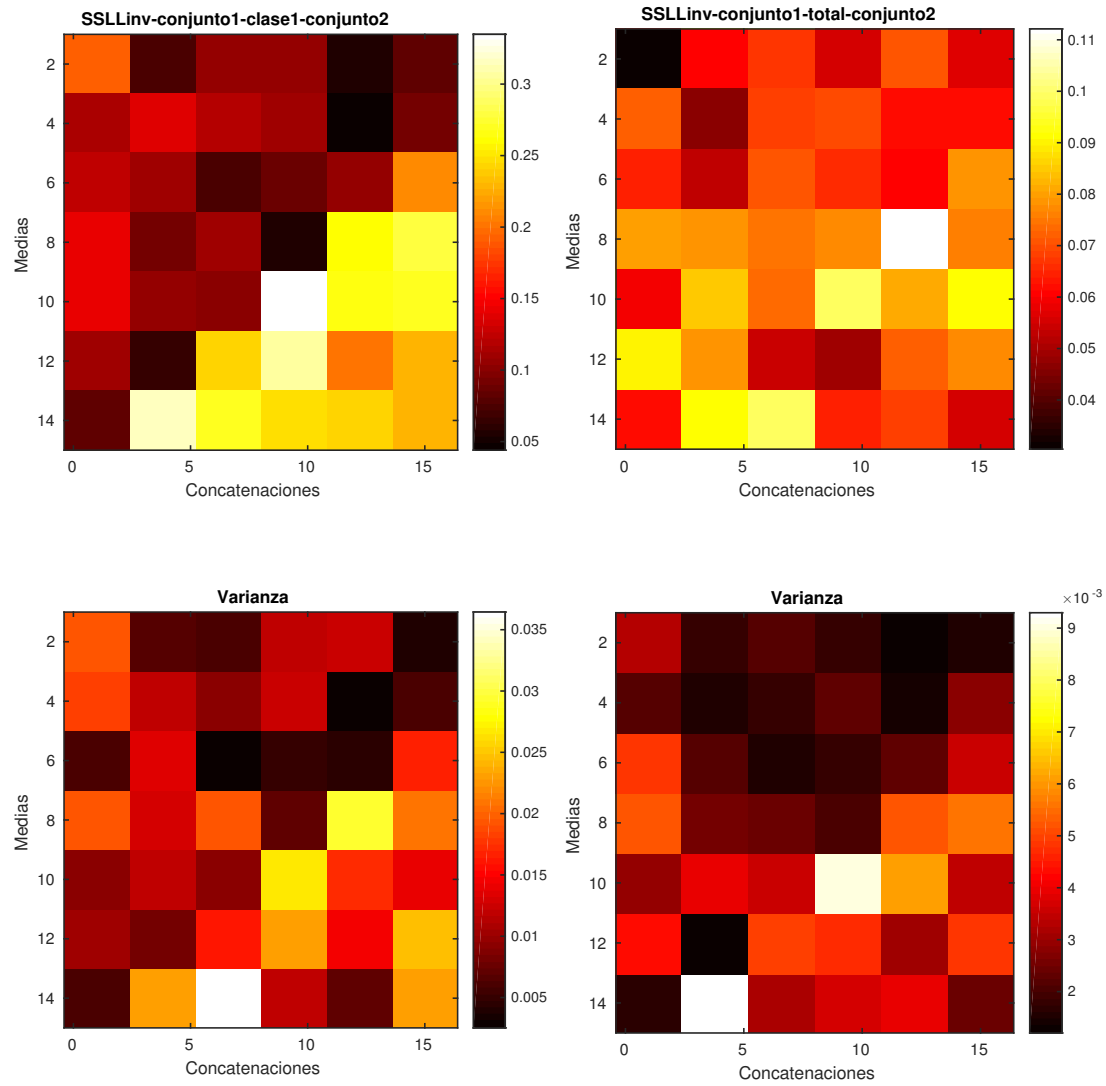


Figura A.46: Mapa de color del subconjunto 1 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

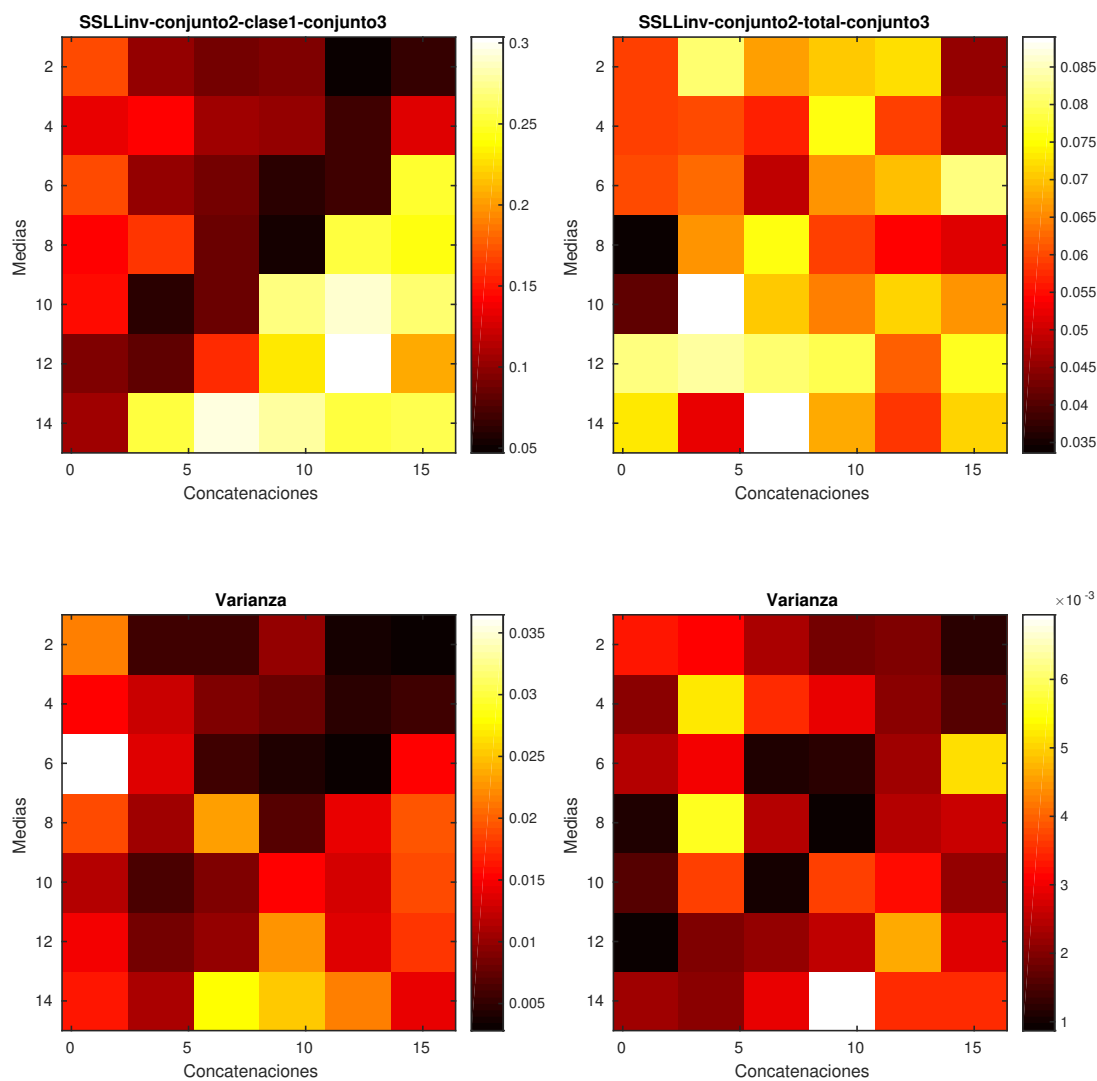


Figura A.47: Mapa de color del subconjunto 2 del método sl. El coeficiente representado a la izquerdam representa la clase del ERP P300, y el de la derecha el total.

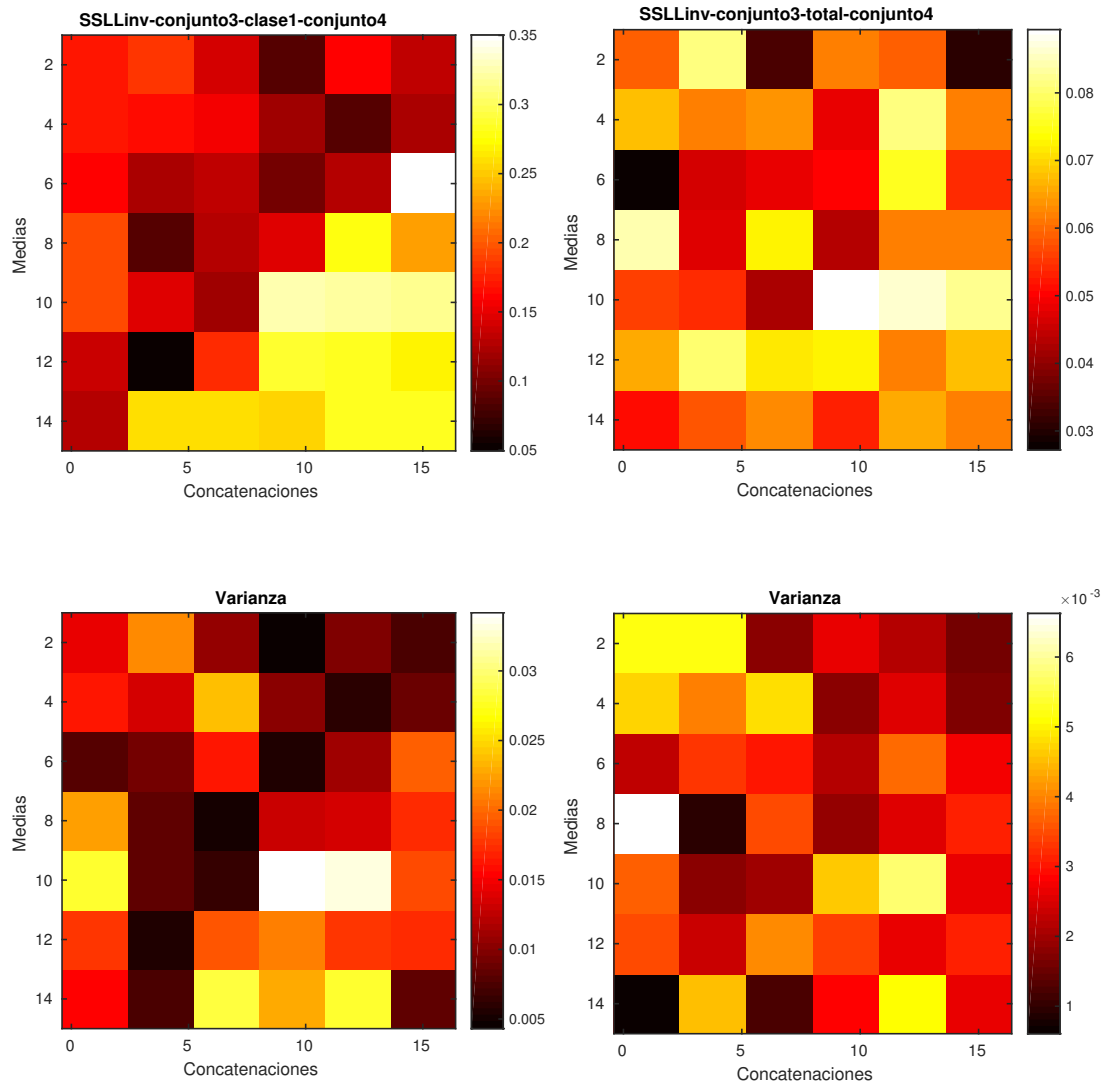


Figura A.48: Mapa de color del subconjunto 3 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

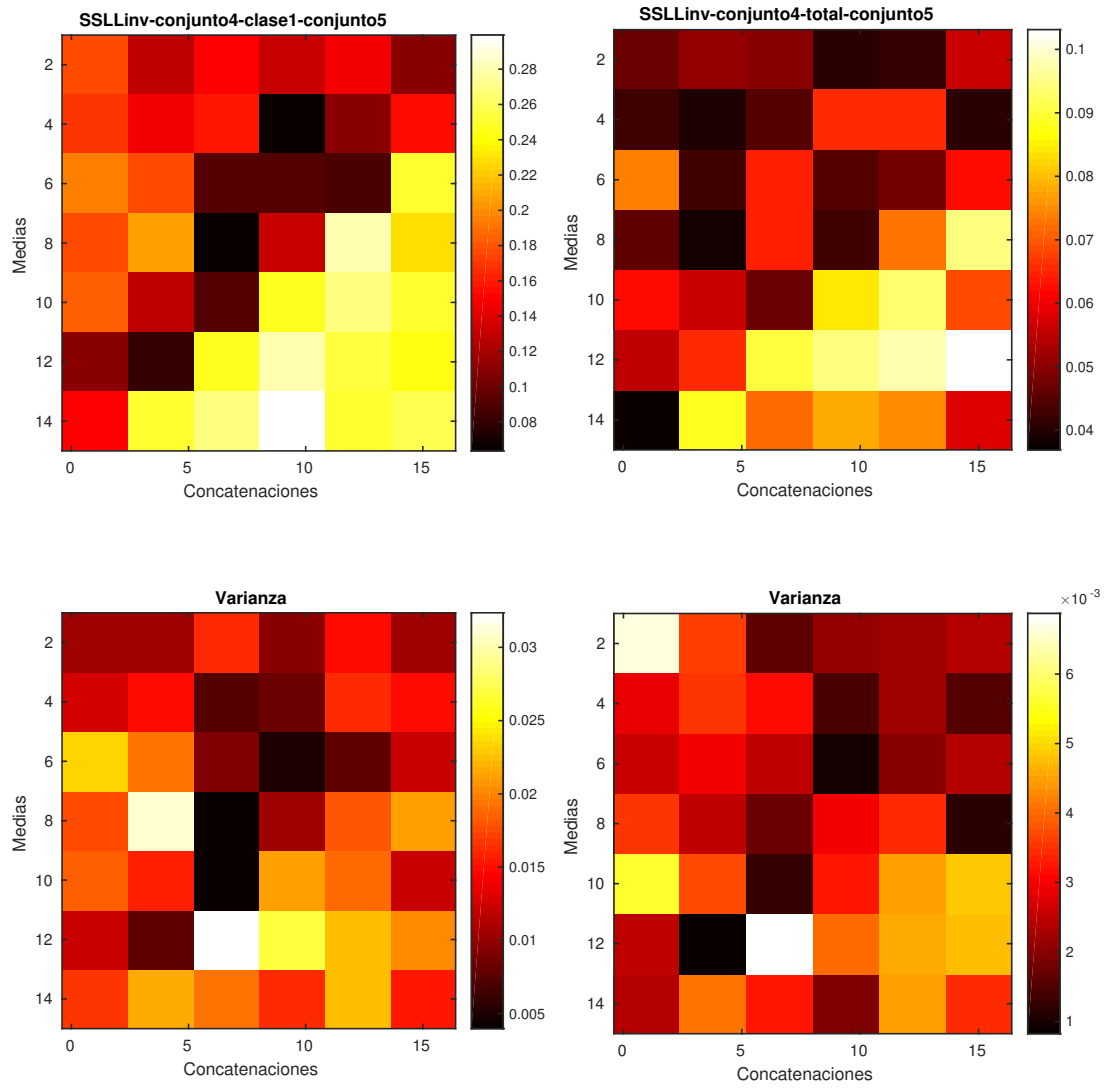


Figura A.49: Mapa de color del subconjunto 4 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.

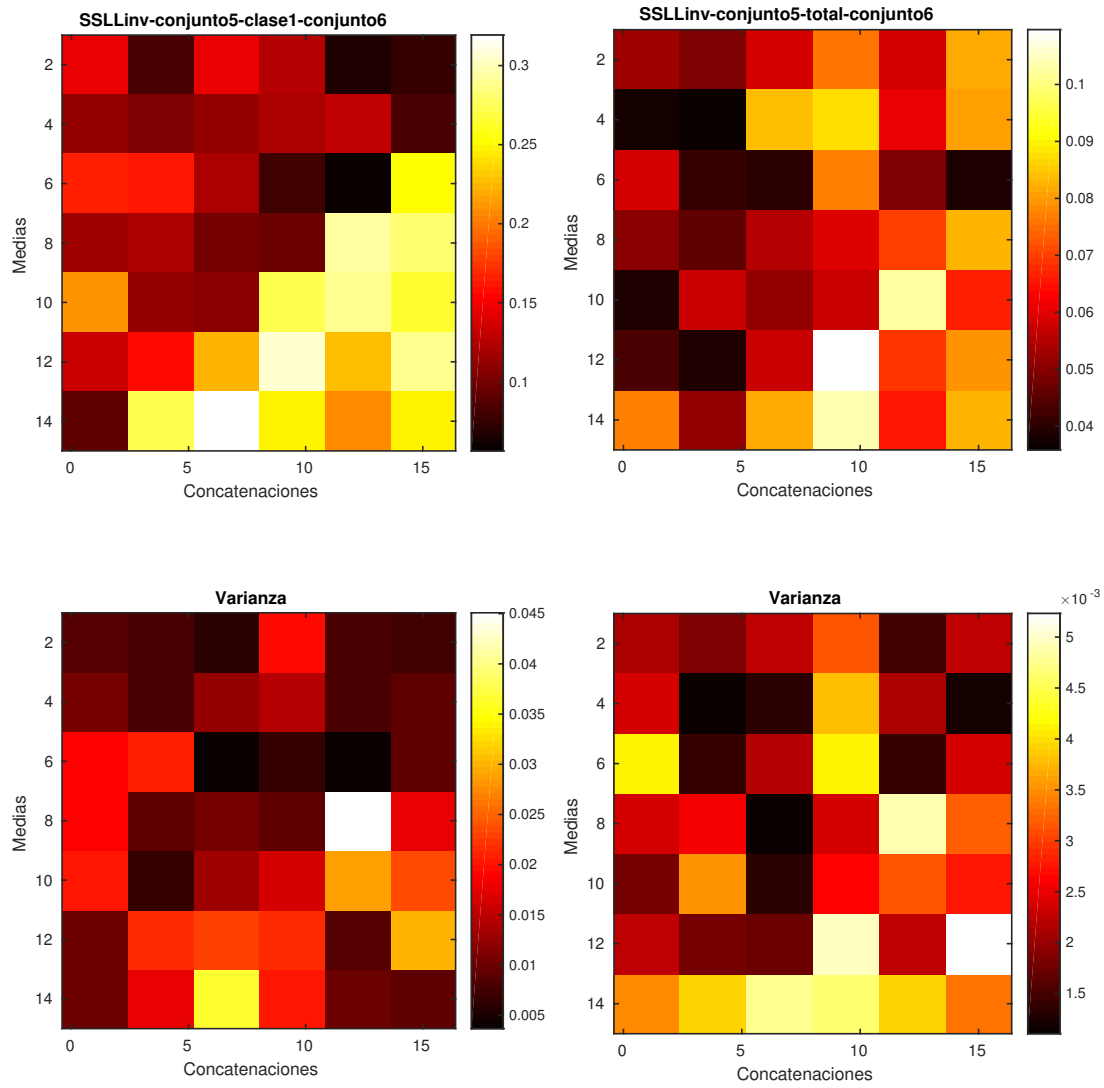


Figura A.50: Mapa de color del subconjunto 5 del método sl. El coeficiente representado a la izquierdam representa la clase del ERP P300, y el de la derecha el total.